

## **GESTIÓN DE RIESGOS INTELIGENTE EN LA ENSEÑANZA DE LA INGENIERÍA DE SOFTWARE**

### **GESTIÓN DE RIESGOS INTELIGENTE EN INGENIERÍA DE SOFTWARE**

AUTORES: Yadira Ruíz Constanten<sup>1</sup>

Lianne Guillén Pérez<sup>2</sup>

Mailín Ochoa Calzadilla<sup>3</sup>

DIRECCIÓN PARA CORRESPONDENCIA: [yadira@cepes.uh.cu](mailto:yadira@cepes.uh.cu)

Fecha de recepción: 21-01-2017

Fecha de aceptación: 28-08-2017

### RESUMEN

El proceso de enseñanza-aprendizaje se encuentra entre las dimensiones más importantes del desarrollo humano en relación con la educación y la cultura. En este proceso el conocimiento es el eje de transformación de la vida teniendo en cuenta el entorno social que rodea al sujeto. La ingeniería de software es una disciplina fundamental en la formación de ingenieros informáticos pues muestra cómo diseñar y conceptualizar un software con calidad, que sea económicamente rentable, mediante el establecimiento y la aplicación de principios sólidos de la ingeniería. En la construcción de sistemas informáticos son muchos los riesgos que deben tenerse en cuenta y de una adecuada gestión depende la mejor manera de mitigar o aprovechar un riesgo o un conjunto de ellos tras la identificación, análisis y respuesta con el propósito de aumentar la probabilidad y el impacto de los eventos positivos y disminuir el de los negativos. El objetivo de este artículo es proponer el uso de una herramienta inteligente para la identificación y

---

<sup>1</sup> Breve resumen curricular: Graduada como Licenciada en Ciencias de la Computación en julio de 2004. Master en Gestión de Proyectos Informáticos, 2007. Posee la categoría docente de profesor Asistente. Se desempeña actualmente como profesora del Grupo de Investigación de Tecnología e Innovación Educativa del Centro de Estudios para el Perfeccionamiento de la Educación Superior de la Universidad de La Habana. Pertenece a la Asociación Nacional de Pedagogos de Cuba y a la Sociedad de Matemática Computación. La Habana, Cuba.

<sup>2</sup> Breve resumen curricular: Graduada como Licenciada en Ciencias de la Computación en julio de 2005. Posee la categoría docente de profesor Asistente. Se desempeña actualmente como Jefa del Departamento de Ingeniería y Gestión de Software de la Facultad 2 de la Universidad de las Ciencias Informáticas. Pertenece al grupo de investigación de Gestión de Proyectos Informáticos y de Ingeniería y Calidad de Software de la Universidad de las Ciencias Informáticas y a la Sociedad de Matemática Computación. lianne@uci.cu.

<sup>3</sup> Breve resumen curricular: Graduada como Ingeniera Industrial en julio de 2004. Posee la categoría docente de profesor Asistente. Se desempeña actualmente como Directora de Capital Humano de la Universidad de las Ciencias Informáticas. Pertenece al grupo de investigación de Gestión de Proyectos Informáticos. Cursó el Diplomado de Administración Pública. mailin@uci.cu.

mitigación de riesgos de software como objeto de aprendizaje en las asignaturas de la disciplina Ingeniería y Gestión de software de la Universidad de las Ciencias Informáticas, en aras de apoyar el proceso de enseñanza/aprendizaje al introducir los contenidos relacionados con la gestión de riesgos en el desarrollo de software.

**PALABRAS CLAVE:** gestión de riesgos; ingeniería de software; inteligencia artificial; riesgo

## **Intelligent risk management in teaching software engineering**

### **ABSTRACT**

The teaching-learning process is among the most important dimensions of human development in relation to education and culture. In this process knowledge is central to the transformation of life considering the social environment surrounding the subject. Software engineering is a fundamental discipline in the training of software engineers because it shows how to design and conceptualize a software quality that is economically viable, through the establishment and application of sound engineering principles. In building computer systems are many risks to be taken into account and appropriate management depends on the best way to mitigate or take advantage of a risk or a set of them after the identification, analysis and response in order to increase the probability and the impact of positive events and decrease the negative. The aim of this paper is to propose the use of an intelligent tool for the identification and mitigation of software risks as an object of learning in the subjects of engineering discipline and management software from the University of Information Sciences, in order to support the teaching / learning by introducing the content related to risk management in software development.

**KEYWORDS:** software engineering; artificial intelligence; risk; risk managements

### **INTRODUCCIÓN:**

Fundamentación de los antecedentes y su pertinencia.

Para gestionar un proyecto de software con éxito, debe comprenderse qué puede ir mal y cómo hacerlo bien (Pressman, 2010), pues durante cualquier etapa de su desarrollo, factores como el entorno tecnológico, los recursos necesarios, las herramientas utilizadas, los requerimientos del cliente y la estabilidad del personal, pueden modificarse sustancialmente y, por tanto, pueden darse consecuencias no previstas inicialmente que alteren su buen

término. Estos eventos que pueden ocurrir o no, ocasionando cambios en los objetivos de un proyecto, son los riesgos. Hay otras definiciones en la literatura: el riesgo es la posibilidad de sufrir una pérdida (SEI, 2012); un evento o una condición que, si ocurre, tiene un efecto positivo o negativo sobre los objetivos de un proyecto (PMI, 2008); la posibilidad de que una amenaza dada impacte las vulnerabilidades de un activo o grupo de activos y cause así, daños a la organización (ISO/IEC 13335-1:2004).

No fue hasta los inicios del milenio, que comenzaron a tenerse en cuenta en definiciones formales, las oportunidades y beneficios que también puede entrañar un riesgo para el proyecto (PMI, 2008) (Kahkonen, 2001) (Mochal, 2002) (Cancelado, 2006) (DACS, 2012) (Bannerman, 2008) (Hall, 2011), lo que suele llamarse riesgo positivo.

Cuando se considera el riesgo en el contexto de la gestión de proyectos de software, a pesar de que se han producido amplios debates sobre la definición adecuada (Cancelado, 2006) (Bannerman, 2008) (Hall, 2011) (Bohem y DeMarco, 1997) (Charette, 1989) (Jacobson, Booch y Rumbaugh 2000), hay acuerdo común en que este implica dos dimensiones: incertidumbre como el acontecimiento que caracteriza al riesgo que puede, o no, ocurrir y el efecto en los objetivos cuando el riesgo se convierte en una realidad que tendrá consecuencias para el proyecto.

Cuando se analizan los riesgos es importante cuantificar el nivel de incertidumbre y el grado de pérdidas asociados con cada riesgo. Este trabajo se facilita al considerar diferentes categorías de riesgos. En la Tabla 1 se muestra un resumen de la clasificación de los riesgos relacionados con proyectos de software.

Tabla 1. Clasificación de los riesgos de software

<b>Criterio</b>	<b>Clasificación</b>	<b>Descripción</b>
Nivel de conocimiento (Pressman, 2010)	Conocidos	Basta con una cuidadosa evaluación del plan del proyecto para que sean descubiertos
	Desconocidos	Se extrapolan de la experiencia en proyectos anteriores
	Impredecibles	Pueden ocurrir, pero son extremadamente difíciles de identificar por adelantado
Nivel de afectación (Pressman, 2010)	Genéricos	Amenaza potencial para todos los proyectos de software
	Específicos	Relacionados con la tecnología, el personal y el entorno específico del proyecto en cuestión
Según el área	Del proyecto	Amenazan los recursos o al plan del

<b>Criterio</b>	<b>Clasificación</b>	<b>Descripción</b>
que amenazan (Fuente y Lovelle, 2006)		proyecto en general
	Técnicos	Amenazan la calidad y/o el desempeño del software en desarrollo
	Del negocio	Amenazan la viabilidad del software a construir y a la organización que desarrolla el software
Según su naturaleza (Alberts, 2006)	Especulativos	Dinámicos: que tienen asociadas tanto pérdidas como ganancias
	Puros	Estáticos: Tienen asociadas solo pérdidas potenciales

En esta investigación la Gestión de Riesgos se refiere a los procesos que se encargan de planificar, identificar, analizar, responder al riesgo, seguir y controlar las actividades planificadas al respecto (PMI, 2008). La investigación en la gestión de riesgos en el ámbito del software procura formalizar conocimiento orientado a minimizar y/o evitar los riesgos, mediante la generación de principios y buenas prácticas de aplicación realista (Ropponen y Lyytinen, 2000). Hasta el momento se han propuesto y utilizado diferentes perspectivas de gestión de riesgos desde que Boehm (1988) atrajo a la comunidad de Ingeniería del Software hacia esta rama. Sin embargo, es evidente que pocas organizaciones utilizan todavía de una forma explícita y sistemática métodos específicos para gestionar los riesgos en sus proyectos software software (Estévez y Pastor, 2005) (Gómez, Hernán, Donoso y Herrera, 2010) (Harvard Business Review Analytic Services, 2012).

Este artículo propone utilizar la herramienta informática Sistema Inteligente de Mitigación de Riesgos como objeto de aprendizaje para las asignaturas Ingeniería y Gestión de Software al impartir los temas relacionados con la gestión de riesgos. Se considera que su uso será de gran importancia pues podrá demostrar a los estudiantes lo complejo que es generalmente este proceso en el desarrollo de un proyecto de software.

## DESARROLLO:

### 1. Ingeniería de Software en la Universidad de las Ciencias Informáticas

La ingeniería de software es una disciplina que circunscribe en ella todos los aspectos de la producción de software (Sommerville, 2006). Pressman (2002) considera que la Ingeniería de Software es el marco que comprende todos los aspectos de este proceso e integra entre dichos factores: un conjunto de métodos, técnicas y herramientas de software, entre otros, con

el fin de desarrollar productos nuevos o mejorar los existentes, por lo que uno de sus objetivos ha sido establecer directrices y principios que sean sistemáticos, predecibles y repetibles a fin de mejorar la productividad y la calidad.

En la Universidad de las Ciencias Informáticas (UCI) la disciplina Ingeniería y Gestión de Software está integrada en su curriculum base por las asignaturas Sistemas de Bases de Datos, Ingeniería de Software 1, Ingeniería de Software 2 y Gestión de Software. Esta es una disciplina fundamental en la formación de ingenieros para el desarrollo de software pues muestra cómo diseñar y conceptualizar un sistema informático con calidad, que sea económicamente rentable, mediante el establecimiento y la aplicación de principios sólidos de la ingeniería. El objeto de estudio de esta disciplina es la proyección, planificación y desarrollo de sistemas informáticos, abarcando todo el ciclo de vida de un software, utilizando técnicas que mejoren la productividad de los diseñadores, incrementen la calidad del producto de software, garanticen un control de todo el proceso de proyección, desarrollo e implantación logrando un funcionamiento de excelencia de la empresa de software.

La asignatura Ingeniería de Software 1 aborda las temáticas: proceso de software, ingeniería de requisitos, estimación y planificación de software y análisis y modelado. Entre las habilidades que debe desarrollar en el estudiante se encuentran:

- Definir el proceso de desarrollo de software para un proyecto.
- Identificar y describir los requisitos a diferentes niveles para una solución informática.
- Administrar requisitos en una solución informática mediante las técnicas y herramientas adecuadas.
- Modelar el entorno de una organización a partir del análisis de sus necesidades.
- Estimar el esfuerzo, el tiempo y los recursos necesarios para desarrollar un software.
- Modelar el funcionamiento del sistema a partir del análisis de los requisitos identificados.
- Modelar la estructura y el comportamiento de un sistema a partir del análisis de sus requisitos.

Por su parte Ingeniería de Software 2 introduce los contenidos relacionados con arquitectura y diseño de software y verificación y validación de software. Las habilidades que permite desarrollar son:

- Describir y especificar la arquitectura de un sistema a desarrollar.
- Modelar el diseño del sistema que se desarrolla.
- Definir y describir un plan de prueba para el proyecto de desarrollo de software.

- Analizar los resultados de la aplicación de una técnica de verificación y validación de software.

Estas asignaturas se complementan con Gestión de Software que profundiza en la evolución y el mantenimiento, la calidad y la gestión de software en aras de que el estudiante sea capaz de ejecutar con eficiencia y eficacia las tareas de la gestión de software, aplicando los principios, métodos y técnicas que garanticen la producción de software con calidad teniendo en cuenta la satisfacción del cliente como meta. Las habilidades a desarrollar por los estudiantes son:

- Caracterizar los procesos implicados en la evolución del software.
- Aplicar los elementos técnicos y de gestión para los procesos de mantenimiento de software.
- Aplicar los procedimientos para la planificación, seguimiento y control de proyectos.
- Aplicar técnicas modernas para la organización de empresas, proyectos, equipos de trabajo y el tiempo de cada desarrollador.
- Aplicar técnicas para los procesos de gestión de riesgo y de configuración y cambios.
- Identificar buenas prácticas para una correcta aplicación del plan de aseguramiento de la calidad.
- Interpretar métricas de calidad de software para los distintos flujos de trabajo del proceso.

El sistema de conocimientos la asignatura Gestión de Software incluye en la unidad didáctica gestión de software elementos sobre la planificación, organización y monitoreo de todas las fases del ciclo de vida de un software. Se estudian los conceptos de gestión de proyectos, modelos clásicos de gestión, roles en la gestión de proyectos, estructuras de gestión organizacional y los tipos de gestión de software. Aborda el control de los cambios, monitoreo y reporte, mediciones y análisis de resultados, corrección y recuperación, recompensa y disciplina, estándares de desempeño. También incluye como parte de la gestión de configuración de software el control de revisiones, gestión de las liberaciones de software, apoyo con herramientas, procesos de gestión de configuración de software, distribución y copias de seguridad. Además se profundiza en la planificación de proyectos, la estimación de esfuerzo y la distribución de recursos y la gestión de riesgos que brinda herramientas a los estudiantes para la identificación, clasificación y posterior tratamiento de los riesgos. Este contenido se sustenta en las definiciones que hace el Project Management Institute (PMI) (PMI, 2008) para los procesos de la gestión de riesgos que se muestran en la Figura 1.

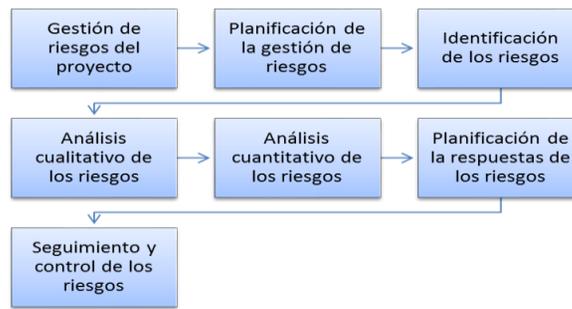


Figura 1. Procesos para la gestión de riesgos según PMI

(Stoneburner, Goguen y Feringa, 2012) afirman que una efectiva gestión de riesgos debe estar totalmente integrada en el ciclo de vida del desarrollo del software y especifica las características de esta integración a través de una tabla (Tabla II) que resume las fases del ciclo, sus características y el soporte que estas necesitan desde las actividades de gestión de riesgos.

Tabla 2. Integración de las actividades de la gestión de riesgos en el ciclo de desarrollo de software.

<b>Fase</b>	<b>Soporte de las actividades de Gestión de Riesgos</b>
Inicio	La identificación es utilizada para apoyar el desarrollo de los requerimientos del sistema, incluyendo los requerimientos de seguridad.
Desarrollo o Adquisición	Los riesgos identificados durante esta fase pueden ser usados para apoyar análisis de seguridad que pueden guiar la arquitectura y el diseño y causar daños durante el desarrollo del sistema.
Implementación	El proceso de gestión de riesgos apoya la evaluación de la implementación del sistema contra los requerimientos.
Operación o Mantenimiento	Las actividades de gestión de riesgos son realizadas para una reautorización o reacreditación periódica, o cuando son desarrollados grandes cambios en el ambiente de producción operacional del sistema.
Cierre	Las actividades de GR son realizadas para componentes del sistema que entrarán en cierre o serán reemplazados, para asegurar que el hardware y el software son adecuadamente manipulados y que la migración del sistema es conducida de manera segura y sistemática.

Se considera relevante instruir a los estudiantes de ingeniería en ciencias informáticas en una adecuada gestión de riesgos pues es la mejor manera de mitigar o aprovechar un riesgo o un conjunto de ellos tras la identificación, análisis y respuesta a lo largo del ciclo de vida del proyecto de desarrollo de software con el propósito de aumentar la probabilidad y el impacto de los eventos positivos y disminuir el de los negativos.

Los modos de actuación del ingeniero en ciencias informáticas están asociados a los procesos del desarrollo de la informatización de la sociedad en sus tres direcciones fundamentales: diagnóstico y transformación de entidades, desarrollo y explotación de sistemas informáticos y diseño y explotación de tecnologías de la información. Al culminar sus estudios universitarios se desempeñarán en los diferentes roles como parte de proyectos informáticos y este conocimiento resulta imprescindible para un buen desempeño.

## 2. Inteligencia Artificial

Una de las definiciones atribuida a Marvin Minsky y que está incluida en (Lasala, 1994), establece que la Inteligencia Artificial (IA) es el arte de construir máquinas capaces de hacer cosas que requerirían inteligencia en el caso de que fueran hechas por seres humanos. Esta ciencia estudia cómo lograr que las máquinas realicen tareas que, por el momento son realizadas mejor por los humanos.

En el momento actual la Inteligencia Artificial se aplica a innumerables actividades humanas. Entre las líneas de investigación más explotadas destacan el razonamiento lógico, la traducción automática y comprensión del lenguaje natural, la robótica, la visión artificial y especialmente, las técnicas de aprendizaje y de ingeniería del conocimiento.

Entre las técnicas de inteligencia artificial existentes se destacan las Redes Neuronales Artificiales que consisten básicamente una serie de unidades de proceso relacionadas mediante unas conexiones ponderadas que están inspiradas en las neuronas de las estructuras nerviosas de los seres vivos; los algoritmos genéticos: es una técnica de búsqueda y optimización basada en los principios de la genética y la selección natural (Araque, Díaz y Gualdrón, 2013); colonia de hormigas: son algoritmos multiagentes diseñados para trabajar en un ambiente con obstáculos y encontrar una solución cercana a la óptima, analizan distribuciones matemáticas, problemas de distribución geográfica y caminos más cortos (Pachón, 2009); lógica difusa: permite trabajar y obtener conclusiones a partir de información de entrada vaga, imprecisa o aparentemente ambigua y sistemas expertos o basados en el conocimiento: son sistemas informáticos que simulan las cadenas de razonamiento que realiza un experto para resolver un problema de su dominio (Li y Sun, 2008). Para conseguirlo, se dota al sistema de un

conjunto de principios o reglas que infieren nuevas evidencias a partir de la información previamente conocida.

La inteligencia artificial ayuda a la toma de decisiones a través de la utilización del conocimiento y la experiencia acumulada, ajustándola a la nueva situación y aprendiendo de cada escenario lo que se considere relevante e incorporando nuevo conocimiento. Este conocimiento pasa a formar parte del sistema, incorporándolo para su posterior utilización en su mecanismo de razonamiento, comparación y aprendizaje.

Según la forma de representar el conocimiento existen diversos tipos de sistemas expertos. Entre ellos se destacan los basados en reglas, los basados en probabilidades y los basados en casos. En esta investigación se propone utilizar un sistema inteligente que utiliza la técnica razonamiento basado en casos para identificar y mitigar riesgos en el desarrollo de software que sea utilizado como un objeto de aprendizaje en las asignaturas de la disciplina Ingeniería y Gestión de Software.

Este tipo de técnica utiliza la madurez de las empresas, los estándares definidos, vivencias anteriores, resultados obtenidos, datos de transacciones de recursos de cualquier tipo, experiencias en el campo donde se quiera implantar, entre otros. Esto permite el empleo de dicho contenido para dar apoyo a la toma de decisiones en la búsqueda de determinados comportamientos, así como resultados que puedan ocurrir en el futuro (Cordero, Ruíz y Torres, 2013).

### 2.1 Técnicas de IA en el proceso de enseñanza/aprendizaje

Sin dudas la educación en estos tiempos es innovadora tras el uso de las tecnologías de la información y las comunicaciones. Su utilización posibilita que sean más efectivos algunos principios pedagógicos y da la posibilidad de explorar nuevos modelos, garantizando la formación a través de la integración multidisciplinar. Para los estudiantes las condiciones para el aprendizaje son más variadas dando un giro a la enseñanza tradicional.

Las tecnologías basadas en inteligencia artificial son de las más demandadas en el proceso de enseñanza/aprendizaje (Sánchez y Lama, 2007). Entre ellas se encuentran los Sistemas Tutores Inteligentes, los Sistemas de Evaluación Automática, los entornos de aprendizaje colaborativo basado en computadora y el aprendizaje basado en juegos. Estos últimos se denominan "juegos serios" para diferenciarlos de los que son destinados al entretenimiento.

En (Sánchez y Lama, 2007) los autores refieren que tal como se plantea en (Brusilovsky y Peylo, 2003) estos sistemas educativos inteligentes están basados en técnicas de inteligencia artificial. Entre las más utilizadas con este fin se encuentran las técnicas de personalización basadas en modelos de estudiantes y grupos, los sistemas basados en agentes inteligentes y las técnicas de web semántica.

### 3. Inteligencia Artificial en la Gestión de Riesgos

Existen diferentes herramientas que permiten, con el uso de técnicas de inteligencia artificial, gestionar los riesgos. La automatización de los procesos relacionados con la gestión de riesgos garantiza la obtención de resultados que apoyan la toma de decisiones en las organizaciones. De este modo es posible desarrollar sus procesos con la mayor calidad posible, en el menor período de tiempo y con la utilización del mínimo de recursos.

Este artículo propone utilizar el Sistema Inteligente de Mitigación de Riesgos como objeto de aprendizaje para las asignaturas Ingeniería y Gestión de Software al impartir los temas relacionados con la gestión de riesgos. Se considera que su uso será de gran importancia pues podrá demostrar a los estudiantes lo complejo que es generalmente este proceso en el desarrollo de un proyecto de software; pues está vinculado a las características de cada proyecto, requiere seguimiento constante y depende de la experiencia acumulada, del conocimiento adquirido y de los estándares definidos. Estos elementos fundamentan que se haya seleccionado la técnica de razonamiento basado en casos.

Esta selección está en correspondencia con una adecuada integración con la gestión de riesgos pues los especialistas en el desarrollo de software poseen el conocimiento explícito y accesible, permitiendo una fácil modificación del mismo. Por otra parte, este sistema inteligente posibilita incorporar nuevas experiencias adquiridas sobre la gestión de riesgos en otros escenarios o problemáticas pertenecientes a los riesgos de proyectos de software en este caso. Las respuestas dadas por este tipo de sistema son consistentes, ya que el conocimiento anterior suministrado a la base de conocimientos puede obtenerse de diferentes expertos o de resultados anteriores, de los cuales se pueden considerar los más adaptables.

La disponibilidad de estas aplicaciones propicia su utilización en el mayor espacio de tiempo posible, facilitando así que, tras la llegada, creación, o concepción de un nuevo proyecto se puedan analizar, identificar y obtener posibles riesgos a incidir en el mismo, contribuyendo a la toma de decisiones. Unido a esto, coexiste la preservación de la experticia y la capacidad para adquirir nuevo conocimiento. De esta forma se perfecciona el que se tiene, se posibilita el aprendizaje de nuevos datos, así como la incorporación de otros riesgos o la variación de la ocurrencia de estos según diferentes escenarios (Cordero, Ruíz y Torres, 2013).

El Sistema Inteligente de Mitigación de Riesgos se basa en la información que introduce el usuario referente a la descripción de un proyecto. Utiliza el Razonamiento Basado en Casos para distinguir por rasgos la descripción dada. Luego los compara con los almacenados en la base de casos con que cuenta el sistema experto y devuelve un grupo de ítems semejantes. A partir de ellos y con el uso de un razonador, que es uno de los componentes de la arquitectura de un Sistema Basado en Casos, devuelve una respuesta que se

asemeja a la obtenida según la similitud del caso entrado con los que se tienen en la base de casos, y a partir de esta información, obtiene las posibles acciones a realizar o estrategias a tomar para mitigar determinado riesgo, ver Figura 2. Finalmente, el sistema almacena el caso si no se encuentra, si es posible obtener información de él, o si puede convertirse en uno nuevo por sus propias características.



Figura 2. Sistema Inteligente de Mitigación de Riesgos

El sistema se desarrolló utilizando la metodología de desarrollo de software ágil para la construcción de sistemas a corto plazo Desarrollo Basado en Funcionalidades / Feature Driven Development (FDD). Se trabajó sobre la plataforma Java, utilizando como lenguaje de programación a Groovy 1.5 con el framework Grails 1.3.5 (creado para este lenguaje). El Entorno Integrado de Desarrollo (IDE) usado es Eclipse, que viabiliza el desarrollo a partir de un conjunto de servicios que tienen su base en varios componentes conectados (plug-in). El sistema tiene niveles de acceso para los roles definidos, garantizando la integridad de la información almacenada. Este mecanismo se basa en el framework de seguridad Grails Spring Security Core 1.0.1. Completa la solución técnica PostgreSQL como sistema gestor de bases de datos objeto-relacional, distribuidos bajo licencia BSD (Berkeley Software Distribution) y con su código fuente disponible libremente.

## CONCLUSIONES:

El desarrollo de sistemas informáticos contribuye al procesamiento de información que tributa como apoyo a la toma de decisiones, lo que implica

que desde los mismos sistemas informáticos se genere conocimiento que puede ser revertido en la transformación de los procesos informatizados. La disciplina ingeniería y gestión de software es un componente fundamental en la formación de estudiantes que formarán parte de la industria de proyectos de software.

Como parte del proceso de enseñanza/aprendizaje el uso de objetos de aprendizaje tiene como beneficios facilitar la comprensión de determinados contenidos. Tal es el caso del uso del Sistema Inteligente de Mitigación de Riesgos que contribuirá a introducir los temas relacionados con la gestión de riesgos en las asignaturas Ingeniería de Software 1 y 2 y Gestión de Software en la Universidad de las Ciencias Informáticas y a lograr que los estudiantes alcancen las habilidades que se definen al utilizar la experiencia que en esta universidad se tiene en el desarrollo de soluciones informáticas.

En la definición de la propuesta se tuvo en cuenta que la complejidad para el desarrollo aplicaciones basadas en el conocimiento para la gestión de riesgos, radica en la forma de representar dicho conocimiento como elemento fundamental. Por las condiciones y características de la Universidad de las Ciencias Informáticas se determinó utilizar como técnica el razonamiento basado en casos.

## BIBLIOGRAFÍA

1. Alberts, Ch. J., (2006). *“Common Elements of Risk”*, Pittsburgh: Carnegie Mellon University, CMU/SEI-2006-TN-014.
2. Araque, J. A., Díaz, J. L., Gualdrón, O. E., (2013). *“Optimización del THD en un convertidor multinivel monofásico usando algoritmos genéticos”*, Revista Colombiana de Tecnologías de Avanzada, Vol. 1, (21), ISSN: 1692-7257, Consultada el 8 de septiembre de 2016, [http://revistas.unipamplona.edu.co/ojs\\_viceinves/index.php/RCTA/article/view/297](http://revistas.unipamplona.edu.co/ojs_viceinves/index.php/RCTA/article/view/297).
3. Bannerman, P., (2008), *“Risk and risk management in software projects: A reassessment”*, Journal of Systems and Software, Vol 8, (12), pp. 2118-2133, Consultado el 10 de septiembre de 2016, <https://pdfs.semanticscholar.org/11c1/086f5a2e6c8a4be9d1f6b9815f8d681acd3b.pdf>.
4. Bohem, B., DeMarco, T., (1997) *“Software Risk Management”*, IEEE Software, Vol. 14, (3), pp 17-19, ISSN: 0740-7459, Consultado el 8 de septiembre de 2016, <http://ieeexplore.ieee.org/document/589225/>.
5. Boehm, B., (1988). *“A Spiral Model of Software Development and Enhancement”*, Computer, s.l.: IEEE Computer Society, Vol. 21, (5), pp. 61-72. ISSN: 0018-9162, Consultado el 10 de septiembre de 2016, <http://dl.acm.org/citation.cfm?id=45801>.
6. Brusilovsky, P., and Peylo, C., (2003). *“Adaptive and Intelligent Web-based Educational Systems”*, International Journal of Artificial Intelligence in Education, Vol. 13, pp.156–169, Consultado el 15 de septiembre de 2016, <http://pitt.edu/~peterb/papers/AIWBEs.pdf>.
7. Cancelado, A., (2006). *“Sistema de administración de riesgos en tecnología informática”*, s.l: IBM Business Global Services.
8. Charette, R., (1989). *“Software Engineering Risk Analysis and Management”*, IEEE Software, s.l: McGrawHill, ISSN: 0070106614.
9. Cordero, D., Ruíz, Y., Torres, Y., (2013). *“Sistema de Razonamiento Basado en Casos para la identificación de riesgos de software”*, Revista Cubana de Ciencias Informáticas, Vol. 7, (2), Abril-Junio, pp 95-112, ISSN: 2227-1899, Consultado el 8 de septiembre de 2016, [http://rci.uci.cu/?journal=rci&page=article&op=view&path\[\]=435&path\[\]=226](http://rci.uci.cu/?journal=rci&page=article&op=view&path[]=435&path[]=226).
10. DACS, (2012). *“Software Acquisition Gold Practice. Formal Risk Management. GoldPractice”*, Consultado el 15 de septiembre de 2016, <https://goldpractice.thesciac.com/practices/frm>.

11. Estévez, J. and Pastor, J. A., (2005). “Implementación y Mejora del método de Gestión de Riesgos del SEI en un proyecto universitario de desarrollo de software”, IEEE Latin American Transactions, Vol. 3, (1). Consultado el 12 de septiembre de 2016, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1468667>.
12. Fuente, A. A. J., Lovelle, J. M. C., (2006). “Proyectos informáticos”, s.l.: Servitec, p. 105, ISBN: 8468972762, 9788468972763.
13. Gómez, R., Hernán Pérez, D., Donoso, Y., Herrera, A., (2010). “Methodology ang Governance of the IT Risk Management”, Facultad de Ingeniería, Universidad de los Andes, Revista de Ingeniería No.31, Enero – Junio, pp. 109-118, ISSN: 0121-499331, Consultado el 30 de agosto de 2016.
14. Hall, D. C., (2011). “Making risk assessments more comparable and repeatable”, Systems Engineering, Vol. 14, (2), pp. 173–17.
15. Harvard Business Review Analytic Services, “Risk Management in a Time of Global Uncertainty”, Zurich Insurance Group, Consultado el 8 de octubre de 2016 de <http://www.zurich.com/internet/main/sitecollectiondocuments/insight/risk-management-in-a-time-of-global-uncertainty.pdf>, 2012.
16. ISO/IEC 13335-1:2004, “Information technology - Security techniques - Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management”, 2004.
17. Jacobson, I., Booch, G., Rumbaugh, J., (2000) “El Proceso Unificado de desarrollo de software”, Madrid: Pearson Education S.A, ISBN: 80-7829-036-2.
18. Kahkonen, K., (2001). “Integration os Risk and Opportunity Thinking in Projects”, London: Fourth European Projects Management Conference, PMI Europe.
19. Lasala, P., (1994). “Introducción a la Inteligencia Artificial y los Sistemas Expertos”, Prensas Universitarias de Zaragoza, Zaragoza.
20. Li, H., and Sun, J., (2008). “Ranking-order case-based reasoning for financial distress prediction”, Knowledge-Based Systems, Vol. 21, 8, pp. 868–878.
21. Mochal, T., (2002). “Facor Positive Risk into Project Planning”, s.l: Tech Republic.
22. Pachón, Á., (2009). “Aplicación de la metáfora de la colonia de hormigas en la administración de direcciones en redes móviles”, Sistemas & Telemática, Vol 7, (13), Enero – Junio, pp. 91-112, ISSN: 1692-5238, Consultado el 20 de septiembre de 2016, <http://www.redalyc.org/articulo.oa?id=411534380003>.
23. Pressman, R., (2002). “Ingeniería del software. Un enfoque práctico”, 5ta ed., McGraw-Hill/Interamericana de España, S.A.U.
24. Pressman, R. S., (2010). “Software Ingneering. A Practitioner’s Approach”. 7th ed. Mc Graw-Hill, ISBN: 978-0-07-337597-7.
25. Project Management Institute (PMI), (2008). “A guide to the Project Management Body of Knowledge”, 4th Ed., s.l.: Project Management Institute, Inc., ISBN: 978-1-933890-72-2.
26. Ropponen, J., and K. Lyytinen, (2000). “Components of Software Development Risk: Hot to address Them?” s.l.: IEEE Transactions on Software Engineering, Vol. 26 pp. 98-111.
27. Sánchez, E. M., y Lama, M., (2007). “Técnicas de la inteligencia artificial aplicadas a la educación”, Revista Iberoamericana de Inteligencia Artificial. Vol. 11 (33), pp. 7-12. ISSN: 1137-3601, Consultado el 2 de septiembre de 2016, <http://polar.lsi.uned.es/revista/index.php/ia/article/viewFile/522/506>.
28. Software Engineering Institute (SEI) | Carnegie Mellon, (2012). “Mission Risk Diagnostic (MRD) Method Description”, Consultado el 28 de septiembre de 2016, <http://www.sei.cmu.edu/library/abstracts/reports/12tn005.cfm>. CMU/SEI-2012-TN-005.
29. Sommerville, I., (2006). “Software Ingneering”, China: Machire Press. 8th edition, ISBN: 978-0-32131-379-9.
30. Stoneburner, G., Goguen, A. and Feringa, A., “Risk Management Guide for Information Technolog y Systems. NIST SP 800-30”. Consultado el 30 de agosto de 2016, <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>, CODEN: NSPUE2, 2012.