

## **Análisis de Métricas de Similitud en Razonamiento Basado en Casos para Administrar Proyectos**

Guillermo Rodríguez, Luis Berdún, Álvaro Soria, Analía Amandi y Marcelo Campo  
ISISTAN Research Institute (CONICET-UNICEN) Campus Universitario, Paraje  
Arroyo Seco, B7001BBO, Tandil, Bs. As.  
{guillermo.rodriguez; luis.berdun; alvaro.soria; analia.amandi;  
marcelo.campo}@isistan.unicen.edu.ar

**Resumen** La administración de proyectos de software es crucial para el desarrollo controlado de los productos de software. Sin embargo, cuando un administrador de proyectos experimentado abandona el proyecto y es reemplazado por alguien con menor experiencia y sin conocimiento del entorno del proyecto, aparece la Amnesia Organizacional. En efecto, administrar un proyecto de software se convierte en una tarea costosa, con incertidumbre y escasa garantía de éxito. En este contexto, Razonamiento Basado en Casos aparece como una técnica adecuada para, por un lado aprovechar la forma de capturar, almacenar y utilizar el conocimiento, y por otro lado, asistir a los administradores novatos en la toma de decisiones cuando se desarrolla la planificación de proyectos. La asistencia al administrador de proyectos se realiza en dos aspectos: (1) asignar recursos a tareas y (2) estimar fechas de finalización de tareas. Esta asistencia se realiza basándose en datos históricos recolectados del uso de la herramienta de administración de proyectos de la organización. Este trabajo presenta un estudio comparativo de distintas métricas de similitud para analizar el desempeño de nuestro enfoque en la recuperación de casos. El estudio se realizó con una base de casos armada con un repositorio de tareas del curso Taller de Ingeniería de Software (Facultad de Ciencias Exactas - UNICEN).

**Palabras claves:** Administración de Proyectos; Razonamiento Basado en Casos; Amnesia Organizacional; Planificación de Proyectos; Agentes Inteligentes.

### **1. Introducción**

La administración de proyectos es una de las principales actividades dentro del desarrollo de software, ya que permite tener visibilidad sobre el alcance del proyecto y tener el control para garantizar la finalización de dicho proyecto [1]. La planificación es una parte esencial de la administración de proyectos; sin un plan efectivo, la posibilidad de fracaso aumenta significativamente para cualquier proyecto. En esencia, el plan es un mapa que muestra cómo ir desde donde uno se encuentra en la actualidad hasta donde se quiera estar. La planificación es la disposición sistemática de tareas para el logro de un objetivo, mientras que el plan establece lo que se necesita lograr y cómo se debe lograr [2].

A medida que las organizaciones han reconocido la criticidad que tiene en la planificación de proyectos, el uso de técnicas de administración de proyectos ha sido el foco en el cual se aplican esfuerzos para contribuir a la mejora de la misma. Cada día más organizaciones han adoptado la administración de proyectos como una estrategia clave para continuar siendo competitivos [3].

Existen varias herramientas<sup>1</sup> de administración de proyectos para asistir a las organizaciones en visualizar el diseño preliminar del plan, reportar el estado actual del proyecto, identificar caminos críticos, proveer información de costos y la duración total del proyecto. Sin embargo, la mayoría de estas herramientas fallan a la hora de dar soporte a los procesos de toma de decisiones y no ofrecen asistencia en cuanto a la captura de conocimiento y la planificación de proyectos. Esto último se debe a que la mayoría se dedican al seguimiento de un único proyecto y no se detienen en recopilar experiencias de proyectos anteriores, desaprovechando todo el conocimiento previamente adquirido. Normalmente, esta experiencia es sólo adquirida por el equipo de desarrollo y cuando éste es desarmado se lo lleva consigo, perjudicando a la organización [8] y derivando en Amnesia Organizacional (AO) [15].

Para atacar este problema, es necesario capturar el conocimiento para luego reutilizarlo en nuevo contexto de tareas y recursos. En efecto, resulta natural aplicar Razonamiento Basado en Casos (*Case-Based Reasoning, CBR*) [16] para dar soporte a estas necesidades. Básicamente, CBR es un técnica de Inteligencia Artificial (IA) que permite resolver un problema nuevo recordando una situación similar previa y reutilizando su información y conocimiento. Otra característica considerada fue el aprendizaje incremental de la técnica de CBR; de esta manera, a partir de nuevos casos que fueron sugeridos existosamente, se puede ir incrementando la información que se posee sobre el dominio de aplicación.

Particularmente, CBR ha sido ampliamente usado para la estimación de costos de proyectos de software. La estimación basada en conocimiento previo es una de las técnicas más atractivas en estimación de costos. En este campo, Wu y otros han estudiado el uso de CBR en combinación con *Particle Swarm Optimization* para estimar el costo de proyectos de software, alcanzado resultados prometedores [4]. Kim ha abordado la incertidumbre en la estimación de costos mediante CBR y *Analytic Hierarchy Process* [5]. Simultáneamente, CBR ha sido aplicado para identificar riesgos y mitigarlos a lo largo de un proyecto de software [6]. Siguiendo la misma línea, Fan y otros han propuesto a CBR como una técnica adecuada para generar estrategias preventivas y controladoras de riesgos en un proyecto de subterráneos de China [7]. Stamelos estableció un enfoque basado en CBR, entre otras técnicas de IA, para capturar, almacenar y propagar y evitar anti-patrones en el manejo de proyectos de software, con el objetivo de educar a futuros managers de proyectos [8]. En esta línea, Koo y otros han propuesto un modelo de CBR para predecir la duración y el costo de un proyecto en su etapa temprana en el contexto de construcción de viviendas, combinando análisis de regresión múltiple, redes neuronales artificiales, algoritmos genéticos, y simulación Monte-Carlo [9]. Jin y otros han trabajado sobre la mejora en modelos de predicción de costos en el contexto de la construcción, mediante la combinación de CBR y regresión lineal múltiple [10]. Siguiendo en el campo de la construc-

---

<sup>1</sup> *Microsoft Project*: <https://products.office.com/en-us/project/project-and-portfolio-management-software>  
*Open Project*: <https://www.openproject.org/>  
*Jira*: [www.atlassian.com/JIRA](http://www.atlassian.com/JIRA)

ción, Ji y otros han abordado CBR, en combinación con algoritmos genéticos, para mejorar la precisión en la estimación de costos en la construcción [11].

Si bien el uso de CBR se ha aplicado satisfactoriamente en el campo de la estimación de costos de proyectos, poco se ha hecho para planificar un proyecto teniendo en cuenta el conocimiento previo basado en el desempeño y aptitudes del recurso humano de la organización. Este problema queda explícito cuando los responsables de administrar los proyectos son novatos y/o carecen de suficiente experiencia en el contexto tecnológico del proyecto. En este punto es donde el uso de CBR puede ser clave para el éxito de un proyecto.

El presente trabajo introduce un enfoque basado en CBR para asistir al administrador del proyecto en la administración y planificación del mismo. El enfoque inicial consta de dos Agentes Inteligentes que asisten al administrador de proyectos. Por un lado, el primer agente recomienda recurso humano para una tarea, teniendo en cuenta el desempeño en tareas anteriores y aptitudes de dicho recurso humano; mientras que el segundo agente estima una fecha de finalización para la tarea que se desea planificar en base al conocimiento histórico del proyecto, los datos de la tarea y del recurso asignado.

Para analizar el desempeño de nuestro enfoque para recuperar casos se realizó un estudio comparativo entre distintas métricas de similitud, las cuales difieren en cómo manipular datos cualitativos y cuantitativos en forma simultánea. Para ello, se utilizó un caso de estudio simulado cuyos datos fueron tomados de un proyecto real del curso Taller de Ingeniería de Software de la Facultad de Ciencias Exactas (UNICEN, Tandil, Bs. As). A través de una serie de ejecuciones, se observó que la métrica de similitud del coseno (*Cosine Similarity*) es la más apropiada para el razonador subyacente de nuestro enfoque. El resto del artículo se organiza de la siguiente manera. La sección 2 introduce el enfoque propuesto y la descripción de la herramienta subyacente. La sección 3 describe el caso de estudio utilizado para evaluar el enfoque. Finalmente, la sección 4 presenta las conclusiones y trabajos futuros.

## 2. Enfoque de Razonamiento Basado en Casos

Para atacar la problemática del manejo eficiente del conocimiento dentro de la organización dada la inexperiencia de los administradores de proyecto novatos, nuestro trabajo propone un enfoque basado en CBR compuesto de dos agentes inteligentes: Agente Recurso y Agente Fecha. Estos agentes van aprendiendo de la experiencia previa de los administradores que han utilizado herramientas de administración de proyectos. De esta manera, se pretende aumentar el conocimiento global de la organización y reducir el impacto que produce la Amnesia Organizacional. La Fig. 1 describe el enfoque propuesto basado en CBR en combinación con Agentes Inteligentes. Siguiendo el modelo propuesto por CBR, el Agente Recurso recibe como entrada un caso nuevo, es decir, una tarea nueva (tarea  $t$ ) que no tiene asignado un recurso humano con habilidades adecuadas, ni tampoco una fecha de finalización. El Agente Recurso **recupera** de la base de casos previos un caso relevante, el cual resulta ser el más similar al caso nuevo luego de la comparación. Este caso relevante se va a usar

como base para asignarle recurso humano y fecha de finalización estimada a la tarea *t* representada por el caso nuevo. En segundo lugar, el Agente Recurso **reutiliza** el caso relevante adaptándolo mediante copia y transformación de atributos. Por ejemplo, una copia posible puede ser asignar el recurso “X” a la tarea nueva, mientras que una transformación posible es la elección del recurso “Y” más similar al recurso “X”, en caso que “X” no esté disponible. Luego, este caso relevante es marcado como resuelto y es **revisado** para verificar que efectivamente resuelve el caso nuevo, convirtiéndose en caso aprendido. En este momento, el Agente Fecha inspecciona el caso aprendido y estima una fecha de finalización para la tarea *t*. Luego, la tarea *t* con un recurso humano asignado (Recurso: “X”) y una fecha de finalización estimada (Fecha: “2015/04/25”) es ofrecida al Administrador del Proyecto quien va asignar esta tarea a un recurso y esperará a que sea ejecutada para luego calificar el desempeño de ese recurso X en la tarea *t*. En nuestro ejemplo se muestra que el caso ejecutado muestra una fecha de finalización real mayor a la estimada (“2015/05/18”) y el desempeño otorgado al recurso “X” por el administrador del proyecto es “7”, y con esta información pasa a ser **recordado** en la base de casos. Como resultado, presentamos un enfoque que guía al administrador de proyectos en la selección del recurso más capacitado para realizar una determinada tarea.

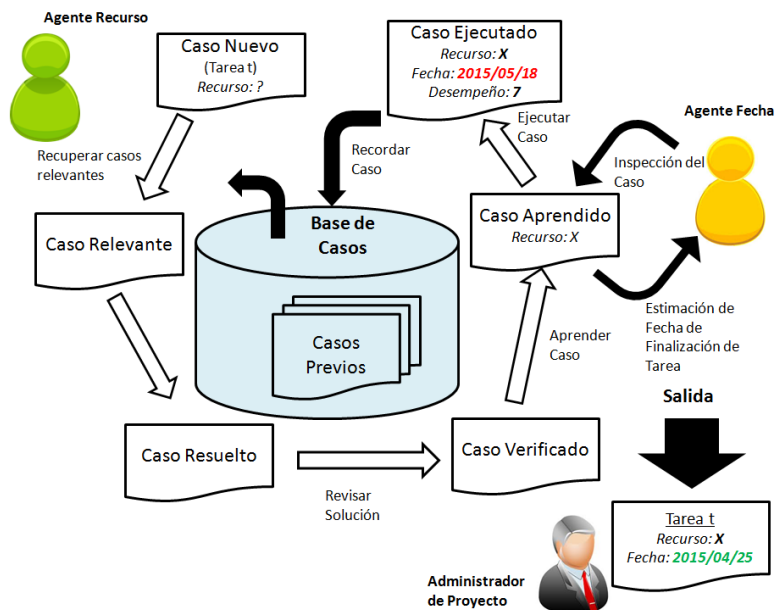


Fig. 1 Resumen del enfoque de CBR propuesto

## 2.1 La Herramienta en Acción

El enfoque presentado está soportado por una herramienta conformada por dos Agentes Inteligentes: Agente Recurso y Agente Fecha. Por un lado, el Agente Recurso es el encargado de realizar recomendaciones de recursos al administrador del proyecto para asignarlo a una tarea, de manera que si éste lo considera apropiado para realizar una tarea puede realizar la asignación correspondiente en la herramienta de

administración de proyectos (por ejemplo, Jira). El agente basa sus decisiones en información histórica sobre las tareas en las que se utilizaron los diferentes recursos que se encuentran disponibles y el desempeño de los mismos en las diferentes tareas. Por otro lado, el Agente Fecha es el encargado de estimar la fecha de finalización que podrá asignar el administrador del proyecto a la tarea. De esta forma, el agente basándose en la información que posee de tareas anteriores, conocimiento del desempeño y habilidades del recurso asignado en otras tareas asignadas anteriormente, y la duración de la tarea del caso aprendido, podrá sugerir una estimación de fecha de finalización de la tarea.

En CBR, un caso es representado por un problema y una solución. En nuestro enfoque, el **problema** está descripto por los siguientes atributos:

- *Categoría de la tarea*: es la clasificación de la tarea entre un conjunto definido de etiquetas, tales como diseño, desarrollo, testing, interfaz de usuario, entre otras. Esta categoría es utilizada como clave de un índice para buscar los casos en la base de casos, acotando el espacio y el tiempo de búsqueda.
- *Skills del recurso*: es una lista de etiquetas que representan las habilidades que posee el recurso humano, tales como conocimientos en Java, Oracle, HTML, Scrum, entre otros, junto con una calificación numérica (de 1 a 10) de dichos conocimientos.
- *Rol del recurso*: es el papel que juega el recurso dentro del proyecto, como por ejemplo diseñador, analista, tester, arquitecto, entre otros.

Por su parte, la **solución** está representada por los siguientes valores:

- *Nombre del recurso*: representa la identidad del recurso humano en forma de nombre y apellido.
- *Fecha de finalización*: es la fecha estimada para terminar una tarea, en función de la duración de la misma.
- *Desempeño del recurso*: es el valor numérico (de 1 a 10) que califica el desempeño del recurso en una tarea.

**Tabla 1 Representación de un caso**

| <b>Problema</b>       |                                |
|-----------------------|--------------------------------|
| Categoría de la tarea | Diseño                         |
| Rol del recurso       | Diseñador                      |
| Skills del recurso    | JavaScript=8, Oracle=3, C# = 9 |
| <b>Solución</b>       |                                |
| Nombre del recurso    | Bernardo Pérez                 |
| Fecha de Finalización | 03-10-2015                     |
| Desempeño del recurso | 9                              |

Siguiendo nuestro enfoque, se puede apreciar que la solución de un caso nuevo es formada por los dos valores que recomiendan los agentes: Nombre del recurso y Fecha de finalización, recomendados por el Agente Recurso y Agente Fecha respectivamente. Por ejemplo, la Tabla 1 muestra la representación de un caso completo. El desempeño es colocado por el administrador del proyecto al finalizar la ejecución de la tarea. Cuando la solución está completa, el caso es recordado para reusarse a futuro.

### 3. Resultados Experimentales

Para realizar el estudio comparativo de métricas de similitud se creó una base de casos con 120 tareas que derivaron de un conjunto de requerimientos funcionales de un proyecto real del curso Taller de Ingeniería de Software. En total se utilizaron 15 recursos, 3 categorías (diseño, desarrollo, testing), 5 roles (analista, arquitecto, diseñador, desarrollador, tester) y 10 diferentes tecnologías para determinar las habilidades de los recursos. Evaluamos la sensibilidad del enfoque explorando distintas métricas de similitud. Este aspecto es de suma importancia ya que va a otorgar fidelidad al enfoque; una mala decisión en este punto afectaría la asistencia.

Las métricas de similitud utilizadas fueron *Dice Sorensen*, *Tanimoto*, *Euclidean Distance*, *Cosine Similarity*, y *Distance Metric*. *Dice Sorensen* se basa en el análisis de la presencia o ausencia de los datos en las muestras. Esta estrategia da menos peso a los datos outliers, manifestando así su utilidad para el análisis de conjuntos de datos heterogéneos [12]. *Tanimoto* es una variante del índice de Jaccard, el cual es un método estadístico para comparar la similitud y diversidad de dos distribuciones [13]. *Euclidean Distance* representa la distancia entre 2 puntos en el espacio dado por la fórmula de Pitágoras. Es una de las distancias más utilizadas para atributos de tipo numérico [12]. *Cosine Similarity* representa coseno del ángulo entre los dos vectores, representado por los atributos de los casos. *Distance Metric* contempla pesos de los atributos y una función que da como resultado 0 ó 1 si los valores a comparar son discretos, o el módulo entre los valores de los atributos en caso de ser continuos [14].

Para cada una de las métricas utilizamos los siguientes indicadores estadísticos: media aritmética ( $\mu$ ), desvío estándar ( $\sigma$ ), mediana (Me), asimetría ( $\alpha$ ) y curtosis ( $\kappa$ ). Estos indicadores fueron aplicados al conjunto de valores de similitud resultante de la comparación entre cada caso de entrada y los casos candidatos almacenados en la base de casos. La asimetría indica que la mayoría de los valores de las métricas de similitud están sesgados a la derecha o a la izquierda de la media aritmética. La curtosis mide el grado de agudeza o achatamiento del conjunto de valores con relación a la distribución normal, es decir, mide cuán puntiaguda es una distribución. Valores cercanos a 0 en asimetría y curtosis muestran que las métricas tienden a una distribución normal. La Tabla 3 muestra 5 casos “nuevos” que fueron dados como entrada al enfoque para analizar cada una de las cinco métricas de similitud. Cada celda de la tabla muestra los valores de los indicadores estadísticos utilizados. La asimetría muestra que la mayoría de las distribuciones están levemente sesgadas a la izquierda ya que sus valores son más bajos que la media aritmética. La curtosis muestra que la mayoría de las métricas tienden a una curva mesocúrtica o normal. Como resultado, *Cosine Similarity* es la métrica más apropiada para seguir explorando nuestro enfoque de CBR ya que maximiza los valores de similitud con bajo desvío estándar, y con valores de curtosis y asimetría que ilustrarían una curva con tendencia normal. Esto nos indica que esta métrica garantiza la estimación de intervalos de confianza para saber cómo se distribuyen los casos recuperados alrededor de la media, permitiendo obtener conocimiento de gran utilidad particularmente en el dominio de proyecto de software.

Tabla 2. Análisis estadístico de las métricas de similitud

| Entrada          | Dice<br>Sorensen   | Tanimoto  | Euclidean<br>Distance  | Cosine<br>Similarity   | Distance<br>Metric   |
|------------------|--|---|--|--|--|
| <i>Entrada_1</i> | $\mu=0.72$<br>$\sigma=0.16$<br>Me=0.76<br>$\alpha=-0.86$<br>$\kappa=0.01$  | $\mu=0.6$<br>$\sigma=0.19$<br>Me=0.63<br>$\alpha=-0.24$<br>$\kappa=-0.83$ | $\mu=0.62$<br>$\sigma=0.16$<br>Me=0.63<br>$\alpha=-0.15$<br>$\kappa=-0.12$ | $\mu=0.76$<br>$\sigma=0.14$<br>Me=0.78<br>$\alpha=-0.66$<br>$\kappa=-0.06$ | $\mu=0.59$<br>$\sigma=0.22$<br>Me=0.6<br>$\alpha=-0.15$<br>$\kappa=-0.97$  |
| <i>Entrada_2</i> | $\mu=0.74$<br>$\sigma=0.14$<br>Me=0.76<br>$\alpha=-0.52$<br>$\kappa=-0.31$ | $\mu=0.58$<br>$\sigma=0.17$<br>Me=0.6<br>$\alpha=-0.41$<br>$\kappa=-0.31$ | $\mu=0.65$<br>$\sigma=0.15$<br>Me=0.64<br>$\alpha=0.07$<br>$\kappa=-0.52$  | $\mu=0.76$<br>$\sigma=0.15$<br>Me=0.8<br>$\alpha=-1.02$<br>$\kappa=0.58$   | $\mu=0.59$<br>$\sigma=0.22$<br>Me=0.55<br>$\alpha=0.02$<br>$\kappa=-0.98$  |
| <i>Entrada_3</i> | $\mu=0.73$<br>$\sigma=0.14$<br>Me=0.76<br>$\alpha=-0.66$<br>$\kappa=0.32$  | $\mu=0.61$<br>$\sigma=0.16$<br>Me=0.6<br>$\alpha=0.09$<br>$\kappa=-0.63$  | $\mu=0.68$<br>$\sigma=0.15$<br>Me=0.7<br>$\alpha=-0.27$<br>$\kappa=-0.67$  | $\mu=0.77$<br>$\sigma=0.11$<br>Me=0.78<br>$\alpha=-0.55$<br>$\kappa=0.17$  | $\mu=0.62$<br>$\sigma=0.23$<br>Me=0.62<br>$\alpha=-0.15$<br>$\kappa=-0.14$ |
| <i>Entrada_4</i> | $\mu=0.72$<br>$\sigma=0.15$<br>Me=0.76<br>$\alpha=-0.79$<br>$\kappa=-0.02$ | $\mu=0.59$<br>$\sigma=0.17$<br>Me=0.6<br>$\alpha=-0.17$<br>$\kappa=-0.46$ | $\mu=0.66$<br>$\sigma=0.18$<br>Me=0.67<br>$\alpha=-0.28$<br>$\kappa=-0.58$ | $\mu=0.78$<br>$\sigma=0.12$<br>Me=0.81<br>$\alpha=-0.9$<br>$\kappa=0.64$   | $\mu=0.62$<br>$\sigma=0.22$<br>Me=0.62<br>$\alpha=-0.08$<br>$\kappa=-1.19$ |
| <i>Entrada_5</i> | $\mu=0.72$<br>$\sigma=0.13$<br>Me=0.73<br>$\alpha=-0.33$<br>$\kappa=-0.48$ | $\mu=0.62$<br>$\sigma=0.18$<br>Me=0.63<br>$\alpha=-0.3$<br>$\kappa=-0.23$ | $\mu=0.64$<br>$\sigma=0.17$<br>Me=0.65<br>$\alpha=-0.21$<br>$\kappa=-0.39$ | $\mu=0.75$<br>$\sigma=0.16$<br>Me=0.77<br>$\alpha=-0.88$<br>$\kappa=0.74$  | $\mu=0.6$<br>$\sigma=0.21$<br>Me=0.61<br>$\alpha=-0.08$<br>$\kappa=-1.05$  |

#### 4. Conclusiones

En este trabajo se presentó un estudio comparativo de métricas de similitud que permite aplicar de manera precisa un enfoque basado en CBR para asistir a un administrador de proyecto en la planificación del mismo, utilizando el conocimiento adquirido por la organización. Los agentes inteligentes Recurso y Fecha contribuyen a disminuir la Amnesia Organizacional que sufren varias organizaciones cuando los administradores de proyectos ya no forman parte de ellas. De este modo, cuando el administrador ya no está presente, el conocimiento histórico aún sigue formando parte de la organización gracias al aprendizaje de ambos agentes inteligentes. Luego de un estudio exploratorio, elegimos *Cosine Similarity* como la métrica más apropiada. Como trabajo futuro, incorporaremos la posibilidad de recomendar múltiples recursos asignados a una tarea para evaluar nuestro enfoque en un contexto de proyectos con tareas complejas y con una granularidad gruesa que no permiten identificar los roles individuales de cada recurso (tareas de alto nivel indivisibles). Asimismo, adiciona-

remos un Agente para coordinar los agentes presentes y recolectar automáticamente métricas de desempeño de los recursos. Una línea interesante a explorar es la purga de los casos que representen anomalías e incluso limitar el envejecimiento de la base de casos. La base de casos puede estar compuesta de casos con un alto valor de longevidad como consecuencia del paso del tiempo y el uso de herramientas de administración de proyectos. Por otro lado, un trabajo derivado es el análisis del reemplazo de un recurso cuando éste deja la organización en medio de un proyecto.

## Referencias

1. Kerzner, H. R. (2009) *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* 10th Ed., John Wiley & Sons, Hoboken, New Jersey, ISBN: 978-1-118-02227-6.
2. Gido, J., Clements, J. (2007) *Administración Exitosa de Proyectos*. Internacional Thomson Editores, México.
3. Kerzner, H. R. (2001) *Strategic Planning for Project Management Using a Project Management Maturity Model*, John Wiley & Sons, USA, ISBN: 0471400394.
4. Wu, D., Li, J. and Liang, Y (2010) Linear combination of multiple case-based reasoning with optimized weight for software effort estimation. *J. of Supercomputing*, 64(3), 898-918, Springer.
5. Kim, S (2013) Hybrid forecasting system based on case-based reasoning and analytic hierarchy process for cost estimation. *J. of Civil Engineering and Management*, 19(1), 86-96, Taylor & Francis.
6. Cordero Morales, D., Ruiz Constanten, Y., Torres Rubio, Y. (2013) Sistema de Razonamiento Basado en Casos para la identificación de riesgos de software. *Revista Cubana de Ciencias Informáticas*, 7 (2), 222-239, ISSN 2227-1899.
7. Fan Z-P., Li, Y-H., Zhang, Y. (2015) Generating project risk response strategies based on CBR: A case study. *Expert Systems with Applications*, 42(6), 2870-2883, Elsevier.
8. Stamelos, I (2010) Software project management anti-patterns. *Journal of Systems & Software*, 83(1), 52-59, Elsevier.
9. Koo, C., Hong, T., Hyun, C. And Koo, K (2010) A CBR-based hybrid model for predicting a construction duration and cost based on project characteristics in multi-family housing projects. *Canadian Journal of Civil Engineering*, 37(5), 739-752, NRC Research Press.
10. Jin, R., Cho, K., Hyun, C., Son, M. (2012) MRA-based revised CBR model for cost prediction in the early stage of construction projects. *Expert Sys. with Apps*, 39(5), 5214-5222, Elsevier.
11. Ji, S., Park, M. And Lee, H-S (2013) Cost estimation model for building projects using case-based reasoning. *Canadian Journal of Civil Engineering*, 38(5), 570-581, NRC Research Press.
12. Deza, M. M. and Deza, E (2009) *Encyclopedia of Distances*, 1-583, Springer.
13. Huang, A (2008) Similarity measures for text document clustering. *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, 49-56, Christchurch, New Zealand.
14. Cunningham, P (2009) A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 21(11), 1532-1543.
15. Othman, R., Hashim, N. A (2004) Typologizing organizational amnesia. *The Learning Organization*, 11(12):273-284, Emerald Insight.
16. Leake, D. B. (1996) *Case-Based Reasoning: Experiences, Lessons and Future Directions*. MIT Press, Cambridge, MA, USA, ISBN: 026262110X.