



Diseño de Sistema para la estimación de esfuerzo en el desarrollo de software utilizando Razonamiento Basado en Casos (RBC) y COCOMO II

Ingrid S. Rincón -UNMSM*

Email: ingrid.rincon@gmail.com

Resumen

Toda gestión de proyecto de software necesita hacer una debida planificación. Existe una gran necesidad de obtener datos objetivos que permiten evaluar, predecir y mejorar la calidad del software, así como el tiempo y coste de desarrollo del mismo. Son puntos importantes en el desarrollo de software y a la vez complicados de calcular o estimar, por ello, en este trabajo se desarrollara un software para la estimación del esfuerzo en el desarrollo de software utilizando Razonamiento Basado en Casos, ya que la estimación de recursos, coste y programa de trabajo para una tarea de ingeniería de software, requiere experiencia, acceso a una buena información histórica; así como las técnicas más usadas y eficaces que existen para la estimación en la planificación de proyectos de software. En este caso se usará el modelo de estimación de COCOMO II.

Palabras clave:

Razonamiento basado en casos, COCOMO II, estimación de proyectos de software.

1. Introducción

Para entender mejor el tema de la estimación podemos compararlo por ejemplo con la construcción de alguna casa, ¿se puede construir sin saber cuánto dinero se va a gastar?, o se podrá construir sin antes saber los pasos a realizar, o ¿cuanto tiempo va a demorar en construirse? Es un hecho que no, por lo que también comenzar a crear un software necesitará una estimación. Un gran error en la estimación de coste puede hacer una diferencia entre beneficio y pérdida.

La estimación del coste de un proyecto en sistemas de información consiste en la aplicación de una serie de técnicas y procedimientos que una organización utiliza para conocer con adelanto el coste que conlleva el análisis, desarrollo, implantación y pruebas del sistema. La estimación precisa de los recursos y tiempo necesarios para el desarrollo de un proyecto, lo cual es esencial para el perfecto desarrollo de cualquier proyecto, pero más si cabe, en el sector informático, en el que los presupuestos y plazos se superan con creces de forma habitual, provocando en muchos casos el fracaso del proyecto.

CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

* Ingrid S. Rincón (Lima, 13 de abril de 1987) Estudiante de X Ciclo de la Facultad de Ingeniería de Sistemas e Informática de la UNMSM, Decana de América.

Con experiencia en labores de desarrollo e implementación de sistemas informáticos en distintas plataformas, ha trabajado para diferentes instituciones y empresas, entre ellas: La Biblioteca Nacional del Perú (Agosto 2007 – Junio 2008), IT Group S.A.C, (Julio 2008-Junio 2009), Altec Soluciones S.R.L (Julio 2009- Actualmente).

Los problemas de la estimación son varios

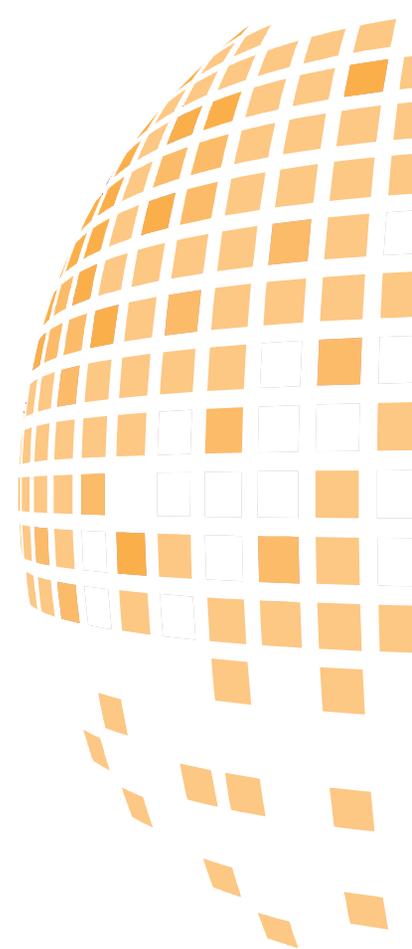
Sabemos que estimar significa, de alguna forma, predecir el futuro, actividad ésta cuya incertidumbre se refleja en la frase “rango aceptable de tolerancia” de la definición dada. Es decir, un primer inconveniente que encontramos es asignar un valor aceptable a cada elemento. Una de las formas de predecir el futuro es tomar en cuenta lo que sucedió en el pasado. En ese sentido, las técnicas de estimación se basan de una forma u otra en datos históricos y experiencias previas sobre elementos o sub-elementos similares al que se debe estimar. Surge de esto un primer problema la identificación de elementos o sub-elementos similares o comparables a lo que debemos estimar.

Las diversas técnicas de estimación buscan entonces reducir el elemento a estimar a la valoración de unidades comunes que permitan, por agregación, cuantificar un sistema. En ese sentido se han definido puntos de función, puntos de casos de uso, clases clave, etc. Ahora bien, la actividad de estimación no se hace una sola vez en el proyecto. A medida que se cuenta con más datos se hacen estimaciones más precisas que nos permiten una mejor planificación de lo que resta del proyecto.

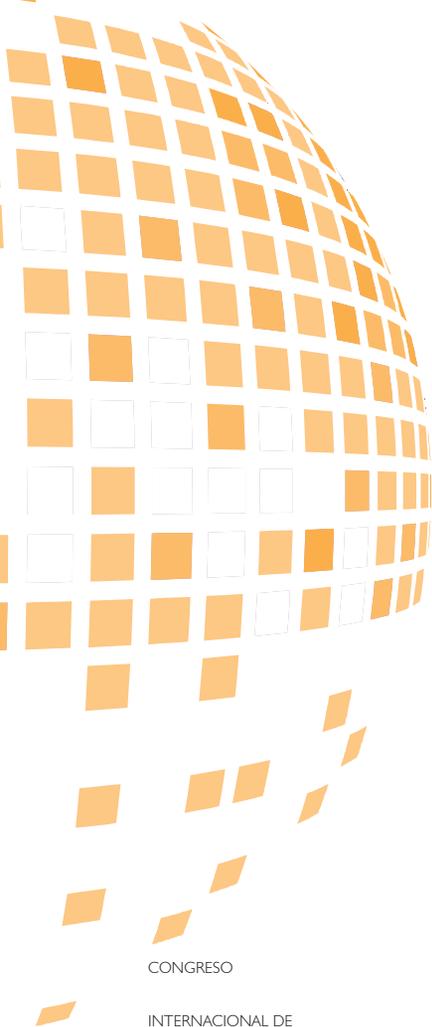
De todos los puntos en los cuales puede realizarse la estimación, cuando menos datos tenemos para hacerla es en el momento inicial, cuando todavía se está evaluando la factibilidad del proyecto. Desde el punto de vista del desarrollo y venta de software específico para terceros esa estimación, que llamaremos temprana, es una de las más importantes.

2. Métodos y modelos de estimación de proyectos de software

Los estudios relacionados con la medición en el nivel de la especificación de requisitos se han centrado fundamentalmente en el desarrollo de métricas para determinar el tamaño y la funcionalidad del software. Entre las de mayor difusión se encuentran las métricas de puntos de función [1], métricas Bang [2] o los puntos objeto [3]. La medición de atributos de calidad de las especificaciones de requisitos del software (ERS) ha sido también objeto de algunos trabajos que van desde la medición de especificaciones formales [hasta la aplicación de métricas para evaluar la calidad de especificaciones expresadas informalmente en lenguaje natural (métricas de facilidad de comprensión del texto contenido en los documentos o métricas de estructura y organización en documentos convencionales y con hipertexto), pasando por técnicas encaminadas a determinar el cumplimiento de los estándares, directrices, especificaciones y procedimientos, que requieren información procedente de revisiones técnicas, inspecciones, Walkthrough, o auditorías[4][5]. La creciente adopción de la tecnología de orientación a objetos en el desarrollo de software ha dado lugar a la aparición de nuevas métricas específicas para este tipo de sistemas.



“ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL”



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

Recientemente se han propuesto métricas para la evaluación de la calidad a partir de modelos producidos en etapas iniciales del ciclo de vida, como son las métricas de calidad y complejidad en modelos OMT, métricas de calidad de los diagramas de clases en UML o las técnicas de medición de modelos conceptuales basados en eventos. La proliferación actual de métricas y la necesidad de medir diferentes aspectos del software está contribuyendo a crear confusión sobre las relaciones entre tales medidas, así como sobre su forma y ámbito de aplicación. Este hecho ha abierto una nueva vía en la investigación orientada hacia la propuesta de modelos, arquitecturas y marcos de referencia (Framework) que permitan la organización de las medidas y la clasificación de las entidades de software susceptibles de medir.

Los Modelos de Estimación [10] son técnicas que identifican contribuyentes claves al esfuerzo, generando fórmulas matemáticas que relacionan estos ítems al esfuerzo. Estas técnicas se pueden aplicar con los siguientes enfoques:

- Bottom -Up: comienza con las partes de menor nivel y provee estimaciones para cada una de ellas.
- Top-Down: estima el producto o proceso completo. Las estimaciones para cada componente son calculadas como porciones relativas del todo.

Cococo (Constructive Cost Model) original[10]

Desarrollado en la década del '70 por Boehm. Revisado con una nueva release en 1995.

Es una colección de tres modelos:

- Básico: aplicable cuando se conoce muy poco del proyecto.
- Intermedio: aplicable luego de la especificación de requerimientos.
- Avanzado: aplicable cuando se termina el diseño.

Todos utilizan la misma fórmula: $E = aS^bF$, donde:

- E: esfuerzo en personas mes.
- S: tamaño medido en KSDI (K-delivered source instructions)
- F: Factor de ajuste (igual a 1 en el modelo básico)
- a, b: s/tablas del modelo en función del tipo de sistema

Modelo de PUTNAM [10]

- Surge en 1978 como solución a un requerimiento de la marina de EEUU para proveer un método para estimar esfuerzo y tiempo. Fue desarrollado por Putnam y lo llamo' modelo SLIM
- Se utiliza para proyectos con más de 70.000 LOC
- Puede ser ajustado para proyectos más pequeños

- Asume que el esfuerzo para proyectos de desarrollo de software es distribuido en forma similar a una colección de curvas de Rayleigh, una para cada una de las actividades principales del desarrollo

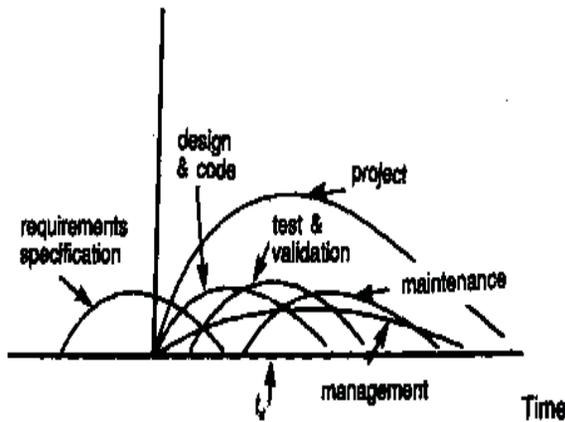


Fig. 1 Modelo De Putnam[10]

Puntos de Función [10]

El método que se está convirtiendo en el estándar de la industria es el definido por el IFPUG, que se llama Function Point Analysis (FPA) y sus autores los definen así:

El método se basa principalmente en la identificación de los componentes del sistema informático en términos de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio. A cada uno de estos componentes les asigna un número de puntos por función basándose en el tipo de componente y su complejidad, y la sumatoria de esto nos da los puntos de función sin ajustar. El ajuste es un paso final basándose en las características generales de todo el sistema informático que se está contando.

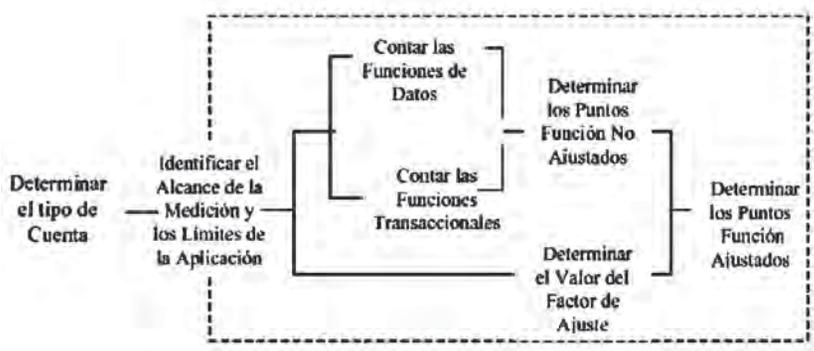
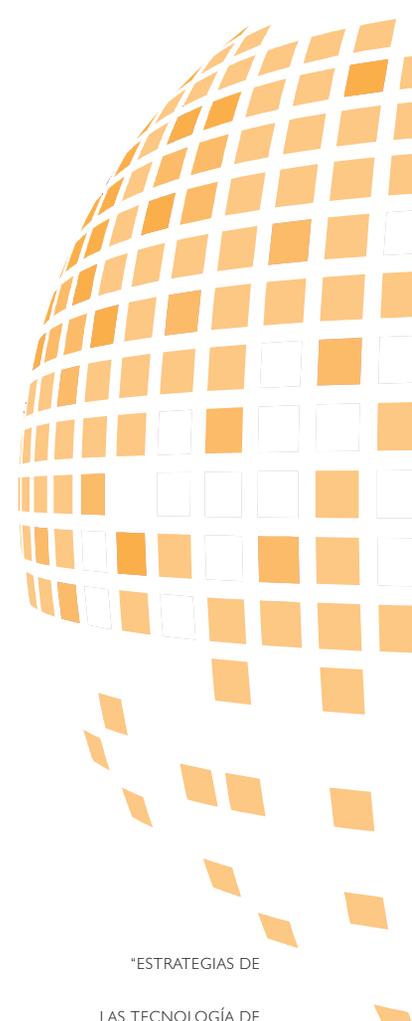
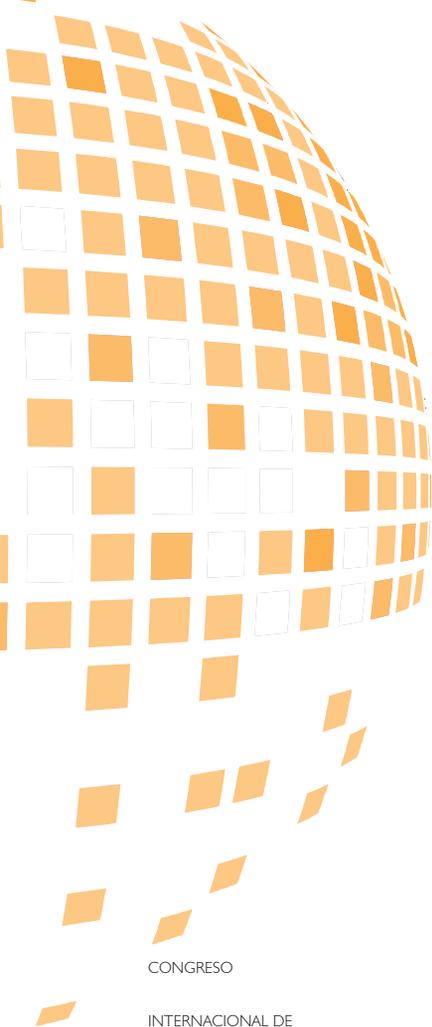


Fig. 2 Pasos del modelo de puntos de función [10]



"ESTRATEGIAS DE
LA TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

Feature Points (puntos de características) [10]

Este método fue propuesto por Caper Jones [7] como una alternativa que permitiera obtener puntos de función en software científico y de ingeniería. Para evitar confusiones con los FP, Jones lo denominó puntos de característica (en inglés feature points). Actualmente es usado con mucho éxito en software del tipo CAD (del inglés Computer Aided Design), sistemas embebidos y sistemas en tiempo real.

COSMIC FFP[10]

A finales de 1998, un grupo de expertos en métricas de software, establecieron el Common Software Measurement International Consortium (COSMIC FFP) [8]. La iniciativa de COSMIC, ha sido básicamente la de dar respuesta a proveedores y a clientes de servicios de desarrollo de software, principalmente en aquellos contratos de terceros donde no había reglas claras acerca del valor de este tipo de servicio. En tal sentido, COSMIC apunta a satisfacer tanto a proveedores de software que deben traducir los requerimientos del cliente en un tamaño del software como un paso clave en la estimación de los costes del proyecto, como a los clientes que quieren conocer ese tamaño recibido como un componente importante para la medición del rendimiento del proveedor. El método se puede aplicar a dominios de software de gestión, tiempo real e híbridos.

3. Evaluación de los métodos de estimación

Feature Points (Puntos De Características)

Es usado con éxito en software CAD. Sistemas embebidos y sistemas en tiempo real.

Cocomo Original

Es una colección de tres modelos:

- Básico: aplicable cuando se conoce muy poco del proyecto.
- Intermedio: aplicable luego de la especificación de requerimientos.
- Avanzado: aplicable cuando se termina el diseño.

MK II FPA

Si se usan herramientas modernas de diseño para el desarrollo del software, y esas herramientas permiten identificar fácilmente las transacciones lógicas, resulta apropiado el uso de este método.

3-D Function Point[10]

Presenta el inconveniente de la necesidad de disponer de mayor cantidad de información acerca del sistema, sobre todo de la complejidad de los algoritmos a implementarse; esta información no siempre está disponible en las primeras etapas del desarrollo.

Full Function Points

Son muy eficientes en la medición de puntos de función en sistemas de control, tiempo real, y embebidos.

Cosmic FFP

El método se puede aplicar a dominios de software de gestión, tiempo real e híbridos.

Puntos De Función

El método que se está convirtiendo en el estándar de la industria es el definido por el IFPUG, que se llama Function Point Analysis (FPA).

Cocomo II

Es la versión mejorada de Cocomo original. Refleja tres etapas en el desarrollo de un sistema

- »1. Prototipos de exploración
- »2. Diseño Inicial (explora arquitecturas)
- »3. Posarquitectura

Como medida usa: Puntos objeto, Puntos de función ó líneas de código fuente. De los modelos vistos anteriormente se puede decir que los métodos que nos dan mediciones necesarias para el cálculo de esfuerzo es COCOMO y Puntos de Función, además de ser las más utilizadas en los proyectos de software.

Resultado

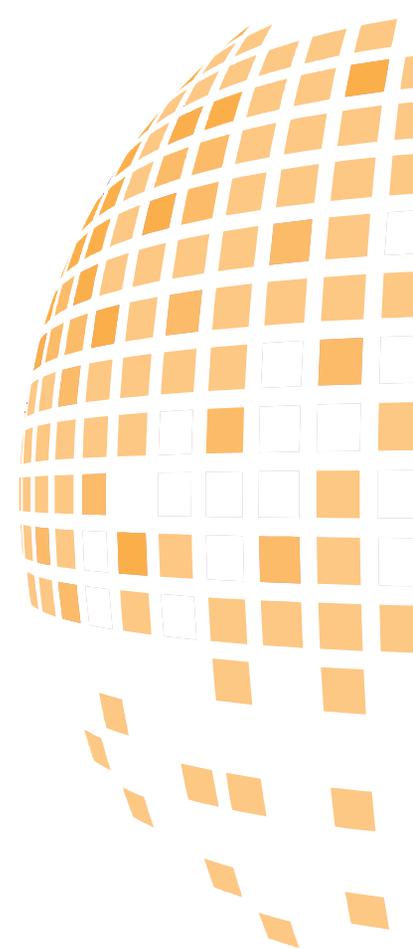
De los modelos vistos anteriormente se puede decir que los métodos que nos dan mediciones necesarias para el cálculo de esfuerzo es COCOMO Y Puntos de Función, además de ser las más utilizadas en los proyectos de software.

Los demás modelos son hechos para cierto tipo de proyectos.

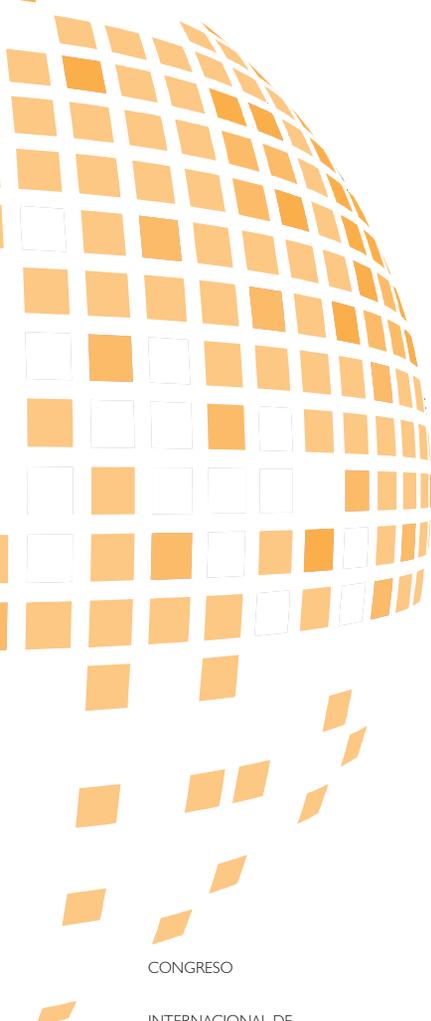
De la evaluación realizada se ha seleccionado como modelo a usar en esta aplicación, pues COCOMO II utiliza puntos de función y puntos de objetos para su cálculo de estimación, además de tener como variables los lenguajes de 3GL, lo cual le da una ventaja competitiva sobre las demás.

4. Razonamiento basado en casos (RBC)

El Razonamiento Basado en Casos, no es más que otro paradigma de resolución de problemas, pero son precisamente las diferencias con el resto de los acercamientos de la inteligencia artificial las que lo hacen tan espe-



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

cial. En lugar de confiar únicamente en el conocimiento general del dominio del problema, o realizar asociaciones a lo largo de relaciones entre descripciones del problema y conclusiones, este paradigma es capaz de utilizar conocimiento específico de experiencias previas, es decir, situaciones de un problema concreto (casos). Ante el planteamiento de un problema no abordado con anterioridad, se intenta localizar un caso pasado similar y adaptar su solución a la situación del problema nuevo. De esta adaptación podemos obtener una nueva experiencia a la hora de resolver problemas con ciertas similitudes, lo que nos lleva a una segunda diferenciación del CBR con respecto al resto de tendencias, el aprendizaje incremental, ya que las nuevas adaptaciones se almacenan como nuevos casos, relacionados, y disponibles para comparaciones futuras.

Modelos y Fases: el ciclo CBR

Los sistemas CBR se descomponen en varias fases que se ejecutan ante la aparición de una nueva situación:

1. Recuperar el caso o casos pasados más similares a la nueva situación presentada.
2. Reutilizar la información y conocimiento de este caso o casos recuperados para resolver el nuevo problema.
3. Revisar la solución propuesta.
4. Recordar o almacenar la nueva solución una vez ha sido confirmada o validada de una manera que pueda ser útil para resolver problemas futuros.

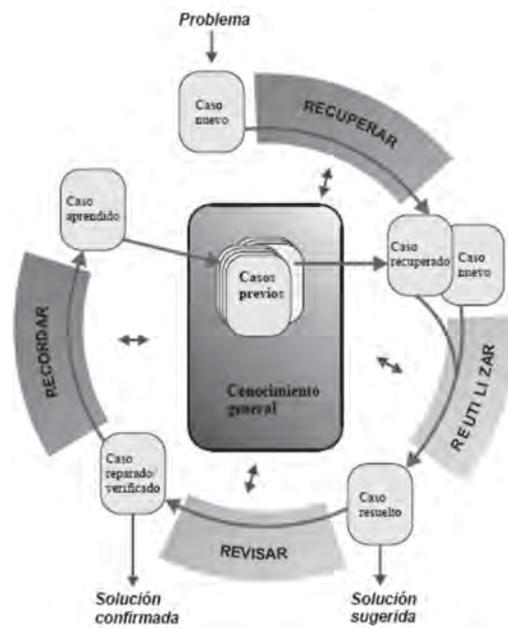


Fig. 3. Ciclo del RBC[11]

Cada uno de estos procesos no son tareas globales, sino que se descomponen en tareas más específicas.

En resumen, un nuevo problema se resuelve recuperando uno o más casos previos (ya experimentados), reutilizando el caso, revisando la solución propuesta, y guardando la nueva experiencia incorporándola a la base de conocimiento existente (base de casos).

Ocurre con cierta frecuencia que tanto las fases de reutilización y de revisión apenas se distinguen por lo que en muchas aplicaciones prácticas, ambas se engloban en una Fase de Adaptación que engloba sendas tareas. Sin embargo no todos los investigadores admiten esta agrupación y es uno de los frentes más discutidos del Razonamiento basado en casos.

5. Planeamiento para la solución

El siguiente esquema nos muestra como funcionará la aplicación desde el requerimiento del sistema que se va a evaluar para estimar su esfuerzo, hasta retroalimentar a la aplicación con el valor real del esfuerzo.

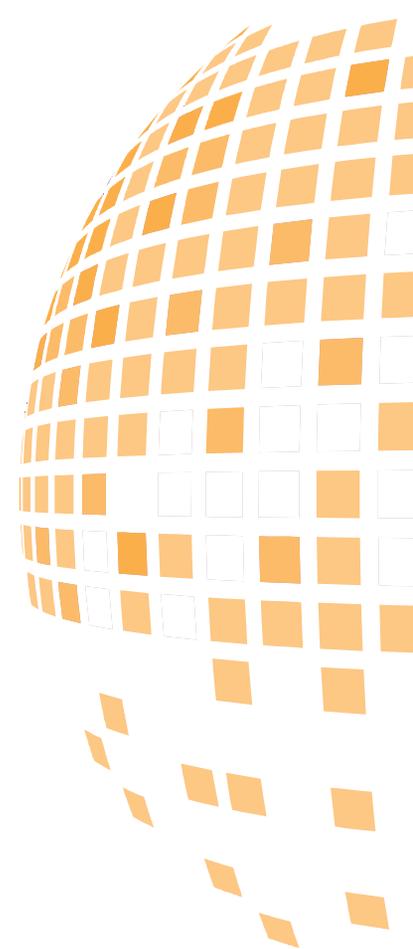


Fig. 4. Modelo conceptual de la solución.

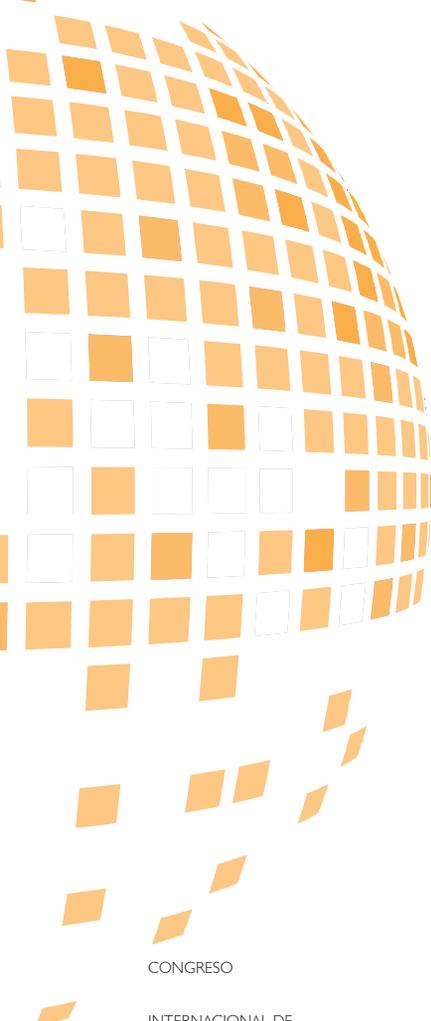
Se considerará para el desarrollo del proyecto los siguientes puntos:

Captura de conocimiento

En este punto se recopilará información sobre el conocimiento para la estimación de esfuerzo. Este proceso recogerá la experiencia acumulada en proyectos pasados. Se plantea recuperar el conocimiento en base a la experiencia o conocimiento acumulado por los jefes de proyectos.



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



Arquitectura

El proyecto presentará una arquitectura que muestra el ciclo de vida del CBR: recuperar, reutilizar, revisar y retener, para lo cual se apoya en el uso de la herramienta ¡Colibri que contiene estos procesos en su estructura y además servirá para validar la solución para el caso de estudio de este trabajo.

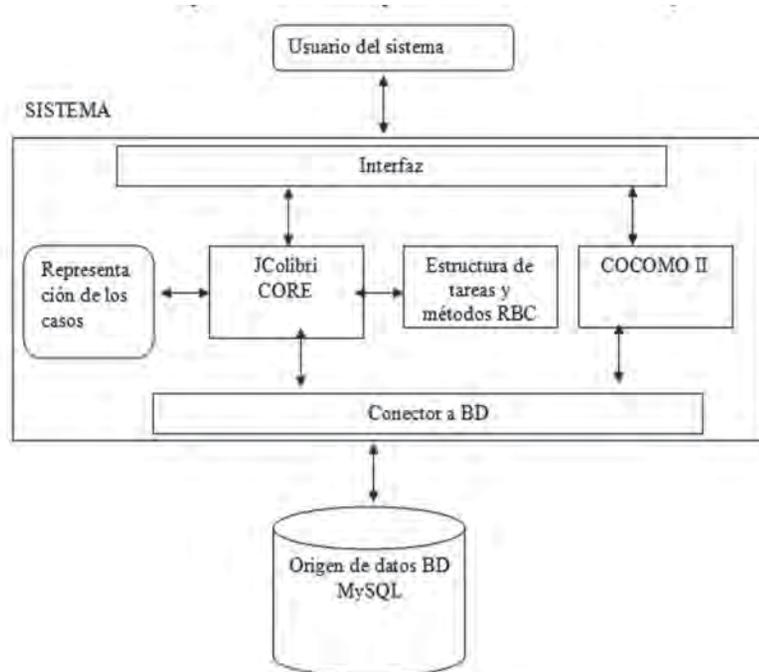


Figura 5. Arquitectura del sistema.

Base de conocimientos

Se creará una base de conocimientos estructura y se tratará de vincular cada caso de la base con las clases correspondientes de ontología, con el fin de realizar una búsqueda eficaz y obtener las mejores soluciones.

Gestión de conocimiento

Para la gestión del conocimiento se definirá una ontología para el caso, la ontología que se creará para esta aplicación empezará estructura y categorizando los casos. Los casos recopilados serán estructurados de acuerdo con las condiciones para configurar, y las situaciones que existan para que cumplan ciertas condiciones. El conocimiento que se recopile para la ontología a definir será aplicable para la estimación de esfuerzo de un proyecto informático.

Herramientas utilizadas

Como herramienta de apoyo se utiliza el ¡Colibri, y el modelo de COCOMO II. Esta aplicación podrá usar parte de la información que se en-

cuenta en una base de casos, de modo que cuando se desee realizar una estimación se busque un caso similar y se le presente al usuario en caso de encontrarlo, los subprocesos que realizará la herramienta ¡Colibri son los siguientes:

- Recuperar los casos similares.
- Reusar la información y el conocimiento.
- Revisar la solución propuesta.
- Registrar en la base de conocimiento, una información derivada de un nuevo caso, como caso nuevo o mejorado.

6. Metodología para la implementación de la memoria en base a casos

Identificar la factibilidad de que la empresa desarrolle su memoria

En esta etapa se debe analizar si la empresa desea generar más conocimiento, la empresa donde se va a implementar esta aplicación deberá desarrollar una memoria organizacional basada en casos, donde registrarán la experiencia en estimación por cada caso (proyectos). La organización debe tener conocimiento documentado, en expertos y se debe compartir con sus miembros.

Identificar el proceso clave

En esta etapa identificaremos el proceso clave de la aplicación: es la estimación de esfuerzo.

Identificar el conocimiento y los expertos

Esta etapa consiste en identificar el conocimiento que se necesite documentar en la memoria organizacional e identificar a un experto que participara en el proceso de obtención de los casos.

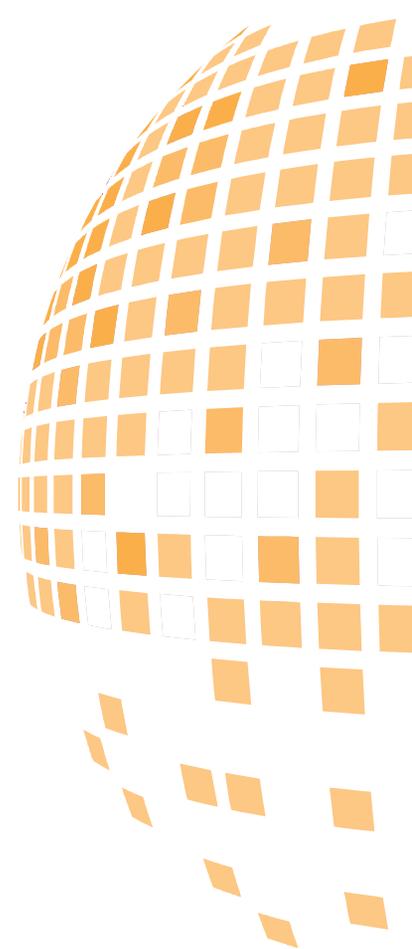
Recopilar y estructurar el conocimiento en forma de caso.

La cuarta etapa consiste en recopilar el conocimiento del experto, y se deberá estructurar los puntos más importantes: el problema, la solución y el resultado.

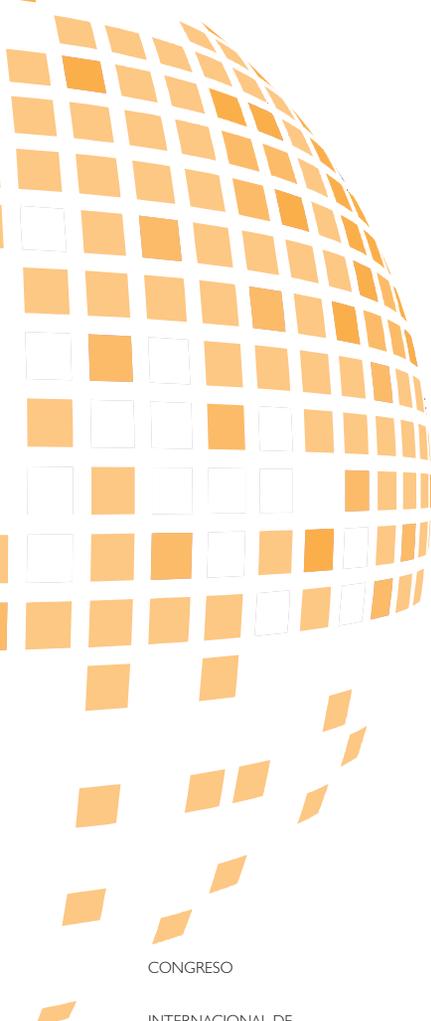
Validar el modelo propuesto

En esta etapa se representa el caso en forma de script para verificar su funcionalidad. Se capturara el caso como parte de la base de conocimiento de la herramienta ¡Colibri.

A continuación se muestra un caso representado en forma de script.



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO
INTERNACIONAL DE
COMPUTACIÓN Y
TELECOMUNICACIONES
COMTEL 2009

Tabla 1 Ejemplo de un Caso Representado en Script

SCRIPT SISTEMA DE FACTURACIÓN	
Nombre	Sistema de facturación
Tipo de caso	Prototipos de exploración
Complejidad de interfaces (Puntos Objeto)	Baja
#Componente 3GL (Puntos Objeto)	0
# de Pantallas por Consulta	2
# de Pantallas para carga de datos	1
Factor de Productividad (Puntos Objeto)	9
Confiabilidad y complejidad del producto	VL: 0.75
Reutilización requerida (Diseño inicial)	LO:0.91
# de Tablas en el sistema	7
# de Reportes	1
# de Interfaz con otro Sistema	0
Nivel de dificultad de la plataforma (Diseño inicial)	NM:1
Experiencia del personal(Diseño inicial)	NM:1
Capacidad del personal (Diseño inicial)	NM:1
Facilidades (Diseño inicial)	NM:1
Duración del proyecto (Diseño inicial-Time)	VH: 1.31
UFP	No calculado aún
KLOC	No calculado aún
Cost Driver	No calculado aún
Flexibilidad del desarrollo	5.07 Riguroso
Manejo de Riesgo y arquitectura	5.65 Arquitectura puede tomarse hasta el 10% del esfuerzo, 5-10 riesgos críticos
Cohesión del equipo de desarrollo	3.29; interacciones básicas cooperativas, objetivos y culturas de accionistas básicamente consistentes, habilidad y disponibilidad básica de accionistas para acomodar objetivos de otros accionistas, poca experiencia previa operando como equipo, visión y compromisos poco compartidos.
Escena 1: Problema	Se desea saber la cantidad de esfuerzo(hombres-mes) que tomara desarrollar el sistema
Escena 2: Solución	La estimación para este tipo de sistemas desarrollado por una persona toma 3 meses, por dos personas toma 1 mes.

Escena 3:
Resultado

La estimación fue correcta con un rango de error de 12%.

7. Modelo para la representación de conocimiento

Este modelo está formado por tres partes principales: Casos, Representación formal de casos y su Representación por medio de herramientas computacionales.

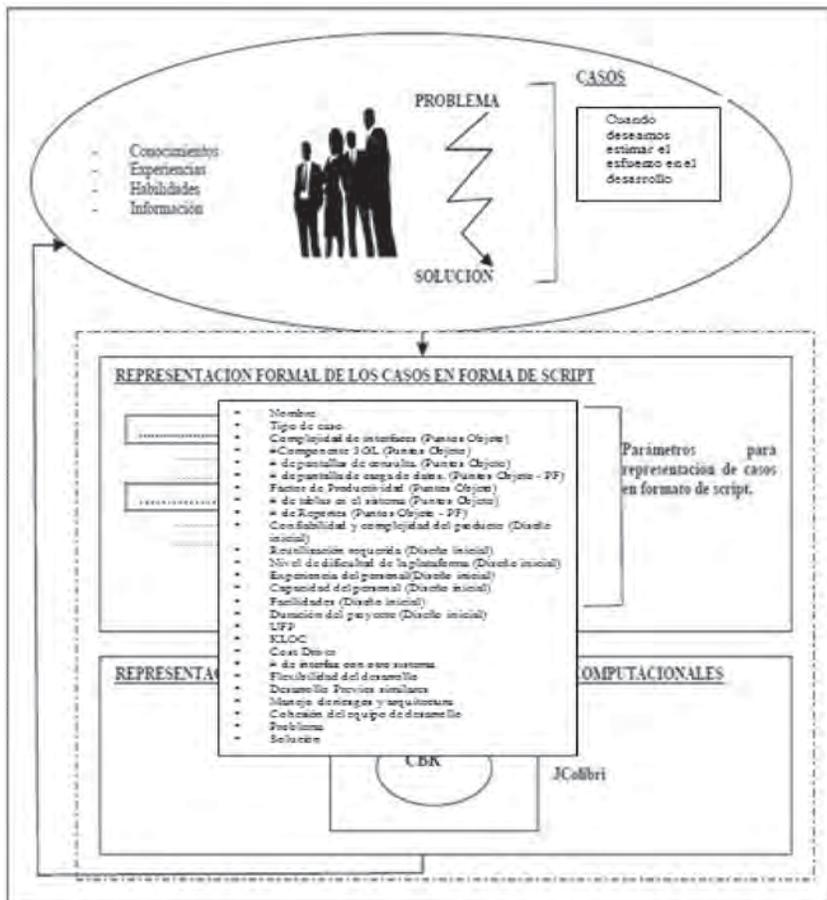
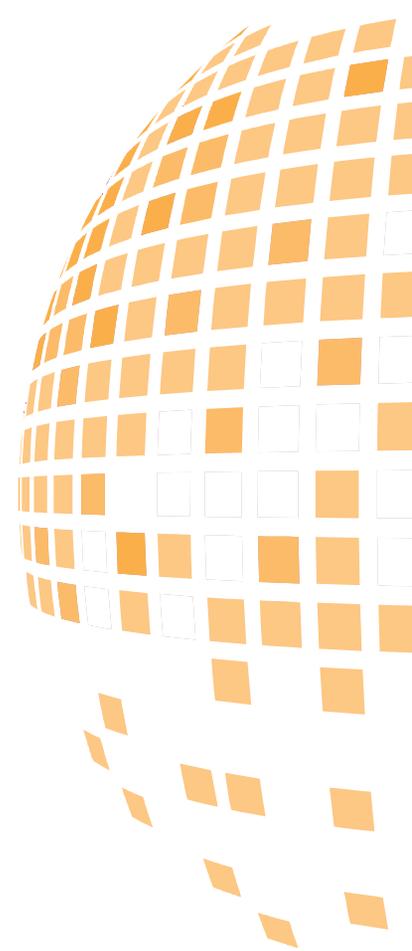


Fig. 6. Modelo de representación de una memoria organizacional.



"ESTRATEGIAS DE
LA TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



Diagrama de actividades

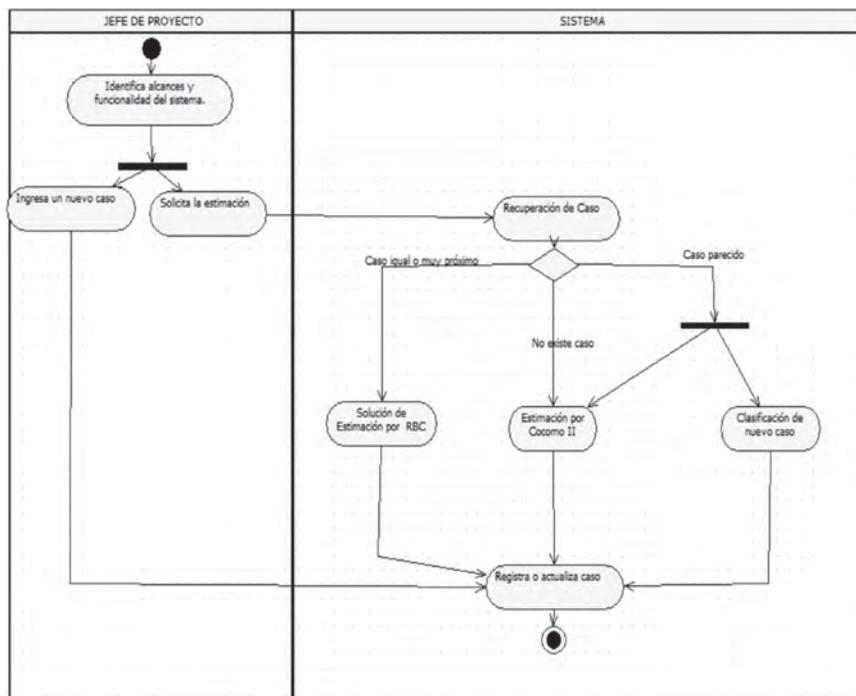


Figura 7. Diagrama de actividades del sistema

8. Conclusiones

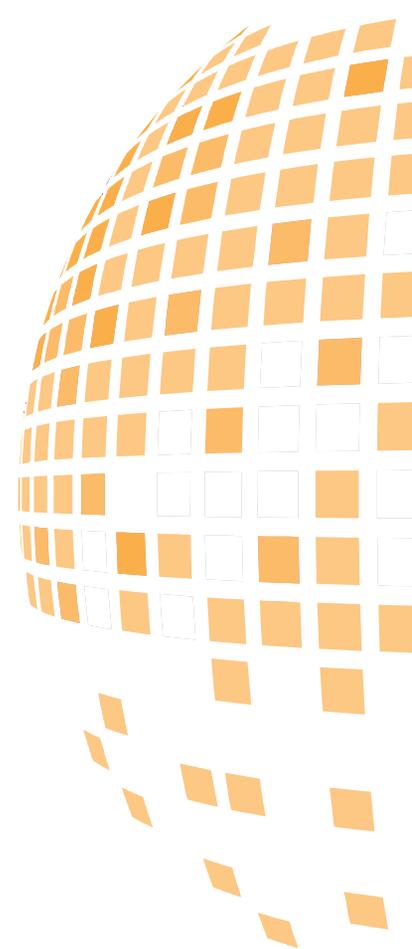
La estimación es una de las primeras tareas en la gestión de proyectos informáticos y se considera una de las más importantes. Ésta es la base para las demás actividades, permite controlar y administrar los recursos que se utilizarán durante el proyecto, sirve como un instrumento muy útil para la toma de decisiones para un responsable de un proyecto de desarrollo de software.

Se puede decir que la estimación no es una ciencia exacta, pues existen muchas variables que influye dentro del proceso de estimación, como variables humanas, técnicas, del entorno y políticas, etc. Sin embargo, es una actividad muy crítica, que se debe realizar con una predicción lo mas exacta posible, en la mayoría de casos esta actividad se realiza solo valiéndose de la experiencia pasada como única guía y esta no es suficiente, los modelos de estimación son muy útiles para esto y es necesario hacer una buena elección de los mismos y realizar constantemente un afinamiento del modo en que se estiman los proyectos.

De lo expuesto anteriormente, se conocen muchos modelos de estimación pero muchos de estos no logran satisfacer las expectativas de los responsables de los proyectos de software, dando lugar a realizar un estudio más profundo sobre las mismas y afinar la manera de dar solución al problema de estimación.

9. Referencias

1. A.J. Albrecht, Measuring application development, Proc. IBM Applications Development Joint
2. SHARE/GUIDE Symposium, Monterey, CA, 83-92, 1979.
3. T. DeMarco, Controlling software projects, Yourdon Press, 1982.
4. B.W. Boehm; B. Clark; E. Horowitz et al., Cost models for future life cycle processes: COCOMO 2.0, Annals Software Engineering 1(1): 1-24, 1995.
5. A. Davis et al., Identifying and measuring quality in a software requirements specification Proc. First International Software Metrics Symposium, Baltimore, 141-152, 1993.
6. B. Farbey, Software Quality metrics: considerations about requirements and requirements specification, Information and Software Technology, 32 (1): 60-64, 1990.
7. Jones, C. 1987. A Short History of Function Points and Feature Points. Software Productivity Research Inc. USA.
8. Symons, Charles R. 1998. Function Point Analysis: Improvements. IEEE Transactions on Software Engineering. Paginas 2-1
9. Antonio A.Sanchez Ruiz Grados, "¡Colibri: estado actual y posibles Mejoras", Universidad Complutense de Madrid 2004-2005.
10. Estimacion de Costos: Problemas y Enfoques Estimacion de Costos. http://cs.uns.edu.ar/~srm/agps/Archivos/Clases/Admc12_4.pdf
11. Razonamiento Basado en Casos: Una Visión General , <http://www.infor.uva.es/~calonso/IAI/TrabajoAlumnos/Razonamiento%20basado%20en%20casos.pdf>



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"