

Círculo de Arquitectos de España

Cuore

CRM



TMS

- El modelado visual de Arquitectura de Software para la empresa según el proceso unificado.
- Estimación en Bases de Datos.
- Soluciones CRM: Visibilidad completa del cliente.
- Teleworking Management System (TMS): La Arquitectura Base en Entornos Distribuidos.

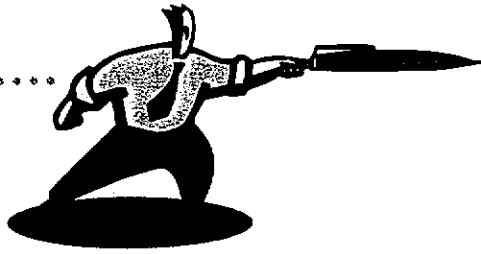
Círculo de **Oracle** de España

cuore

SECRETARÍA DEL CUORE

P.A.P. Congresos
C/ Gil de Ontañón, 21
Telfs.: 91 367 53 65 - 91 377 30 63
Fax: 91 377 46 69
28027 Madrid (España)





Sumario

Pág **6**



SOFTWARE

**El modelado visual
de Arquitectura de Software
para la empresa**

según el proceso unificado

Grady Booch

Pág **14**



DIVULGACIÓN

**Estimación en Bases
de Datos**

Mario G. Piattini Velthuis
Eduardo Fernández-Medina Patón

Pág **22**



ORACLE informa

Soluciones CRM:

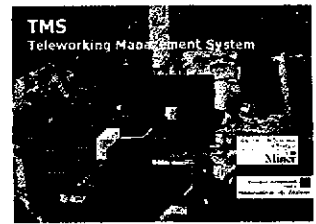
Visibilidad completa del cliente

Antonio Pérez

Pág **26**



TMS



**Teleworking Management
System (TMS)**

*La Arquitectura Base en Entornos
Distribuidos*

Enrique Guzmán

Pág **33**



LECTURAS CUORE



Libro ■ **Fundamentos y Modelos
de Bases de Datos**
2ª Edición

Adoración de Miguel
Mario Piattini Velthuis

Pág 34



SEMINARIOS



X Congreso Nacional de Usuarios de Oracle

Pág 36



PROPUESTAS A ORACLE

El proceso de propuestas a ORACLE de mejoras de productos
Enhancement System

Pág 40



CITAS



David Abreu

Cuore

Responsable:

Pedro Poveda González

Consejo Editorial:

José Ángel Alonso, Rafael Rojo, Pedro Poveda y David Abreu.

Colaboradores:

Arturo Gutiérrez, Javier Gutiérrez, Carlos Antón Martínez, Christopher Adelman, Mario Piattini Velthuis y David Abreu.

Autoedición y Producción:

SIM, S.L. • Tel.: 941 26 38 49

Depósito legal:

M-24195

Impresión:

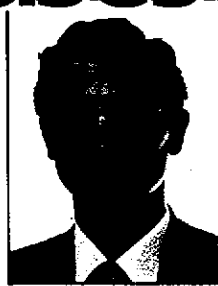
RIOJAGRAFIC, S.A.L.

La Revista CUORE no se hace responsable de las opiniones vertidas por sus colaboradores.

Fe de erratas: En el último número no se publicaron las referencias bibliográficas del artículo "Trampa en el Cyberspacio". Si usted está interesado en conocerlas, consulte la página web de Cuore: www.cuore.es



Estimación en Bases de Datos



MARIO G. PIATTINI VELTHUIS
 Universidad de CASTILLA LA MANCHA
 E-mail: mpiattini@inf-cudm.es



EDUARDO FERNÁNDEZ-MEDINA PATÓN
 Universidad ANTONIO DE NEBRUJA
 E-mail: efmedina@di.unnet.es

En este trabajo se muestra una visión histórica de la estimación del coste y del esfuerzo de sistemas informáticos y se analizan las repercusiones que tienen tales estimaciones para el correcto desarrollo y planificación del sistema. La estimación del esfuerzo de desarrollo de sistemas de información ha sido ampliamente investigada (aunque aun sin encontrar una solución firme e integral), pero el campo de las bases de datos se encuentra falto de métodos de estimación, por lo que en este trabajo se muestran unos métodos para ello.

■ 1 INTRODUCCIÓN

Estimar es predecir o vaticinar el valor que tomará algún parámetro de la construcción de un sistema informático antes de ser construido. Uno de los problemas históricos en la ingeniería del software es la estimación del coste de un producto software. Una práctica habitual en muchas organizaciones software es usar el tamaño de un programa para estimar el coste de desarrollo. Sin embargo, en la mayoría de los casos no es posible tener esa variable al principio del proyecto y ha de ser estimada. El software es el elemento más caro en los sistemas basados en computador y un gran error en la estimación del coste puede marcar las diferencias entre las ganancias y las pérdidas. En realidad son cientos los factores que pueden afectar al coste del

te en el coste software. En general no es posible predecir el coste del sistema antes de construirlo debido a la gran cantidad de factores e incertidumbres que le afecta (Ebrahimi, 1999).

La razón por la que la estimación de esfuerzo y coste de proyectos software ha sido problemática es porque ha de ser hecha al principio del ciclo de vida, momento en el que los valores de muchos factores del desarrollo del software son aún desconocidos (Abdel-Hamid, 1993; Graham, 1995).

A lo largo de la historia ha aparecido un gran número de modelos de estimación. Estos modelos han sido especialmente diseñados para sistemas tradicionales. En la Tabla 1 se muestran algunos de éstos. En Zuse (1998) se puede encontrar una breve historia de los modelos de estimación más importantes.

software. Una aproximación no siempre sencilla es conseguir un modelo lineal o no lineal, paramétrico o no paramétrico que describa la relación entre el coste software y los factores que influyen en éste. Lo que se hace en realidad es tomar solamente los factores que influyen más directamen-

Muchos autores a lo largo de los años 70 analizaron datos de proyectos usando técnicas estadísticas intentando identificar los principales factores que pudiesen contribuir al coste de desarrollo de software. Algunas técnicas se basaron en el tamaño del sistema midiendo las líneas de código fuente, como el modelo COCOMO. Otros modelos se basaron en el análisis de la funcionalidad de los sistemas, como los puntos de función (Albrecht, 1979), la métrica Bang (DeMarco, 1982) y una versión de puntos de función llamada MARK II. Una historia extensa de los modelos de estimación del tamaño funcional de los sistemas puede ser encontrada en el capítulo 2 de Jones (1996).

Ahorra bien, en la ingeniería del software existe un campo de desarrollo, que son las bases de datos, al cual no se le ha prestado demasiada atención en cuanto a la estimación de su creación y explotación. Las técnicas de estimación comentadas en las líneas anteriores hacen referencia al desarrollo de sistemas de información orientados al proceso y no a los datos como es el caso del desarrollo de las bases de datos, por lo tanto tales técnicas de estimación no son directamente utilizables para vaticinar el tiempo de desarrollo de las bases de datos.

Para que un método de estimación de esfuerzo de desarrollo de bases de datos sea bueno, se debe cumplir que sea independiente de la máquina donde será instrumentada la base de datos y del sistema de gestión de bases de datos empleada, aunque debe poder adaptarse a cualquier entorno de desarrollo utilizando datos de dicho entorno y ajustando así el modelo de estimación. Además ha de poder ser aplicado en la etapa de la captura de requisitos y que los resultados de la estimación estén bien correlacionados con el coste o el esfuerzo de desarrollo del sistema. La estimación hecha al comienzo del proyecto no debe ser considerada inalterable. Al contrario, las estimaciones deben ser un proceso continuo mediante constantes modificaciones con los datos coleccionados des-

Nombre del Modelo Organización Año de Aparición

PRICE	RCA	1960
Delphi	Rand Corp.	1966
Nelson's	SDC SDC	1966
Wolverton	TRW	1974
RCA Price-S System	RCA	1976
Halstead		1977
Walston y Félix	IBM	1977
SLIM de Putnam		1978
Método de Puntos de Función		1979
Midelo Parr		1980
Modelo COCOMO	TRW	1981
SOFTCOST JPL		1981
Bailey y Basili	NASA	1981
Métrica Bang		1982
Puntos de Característica		
Puntos de Función MARK II		1988
Pfleeger Model		1989
Modelo COCOMO 2.0		1996

TABLA 1 • Historia de los modelos de estimación de coste

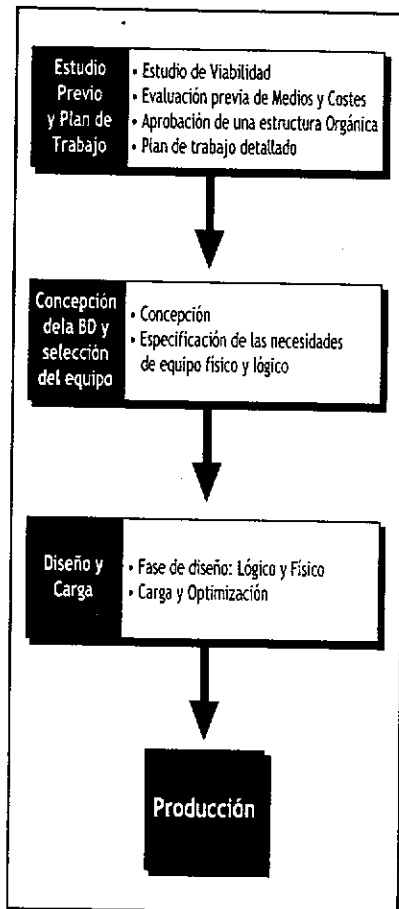


FIGURA 1 • Fases para la puesta en marcha de una Base de Datos

de las actividades de control. Como indica Londeix, "Comparando los datos actuales con los estimados en cada paso, la exactitud de la siguiente estimación puede ser mejorada. En cada paso es necesario hacer posible una mejor predicción futura" (Graham, 1995).

La siguiente sección analiza el ciclo de vida completo de una base de datos para así poder determinar los puntos clave donde pueden ser realizadas las estimaciones, es decir, en qué etapa o etapas (lo más tempranas posibles) se puede disponer de algún producto o algún modelo que pueda ser medido y que a través de esas medidas puedan ser realizadas estimaciones. En posteriores secciones se describen algunos modelos de estimación del esfuerzo de desarrollo de bases de datos.

■ 2 CICLO DE VIDA DE LAS BASES DE DATOS

Al igual que existe un modelo clásico de ciclo de vida en cascada (Piattini, et al., 1996) para el desarrollo de sistemas de información, también se pueden establecer unas etapas o fases secuenciales con realimentación que representen el ciclo de vida de las bases de datos.

La creación de una base de datos no puede improvisarse puesto que suele ser una operación muy difícil, larga y costosa, que ha de ser planificada y gestionada adecuadamente si no se desea tener pérdidas en las organizaciones desarrolladoras. En De Miguel y Piattini (1993) se proponen unas fases para la puesta en marcha de una base de datos, cuyo esquema puede ser observado en la Figura 1.

Las actividades principales que se realizan en cada etapa son las que se describen en el siguiente esquema

Estudio Previo y Plan de Trabajo.

- Estudio de viabilidad. Se concreta la voluntad de los directivos de abordar el proyecto, ya que si no existe un serio compromiso por parte de éstos, existirá mucho riesgo de fracasar, y será conveniente aban-

donar el desarrollo de la base de datos.

- Evaluación previa de medios y costes. Consiste en dar una cantidad aproximada y orientativa del coste global del proyecto y de los recursos de personal a utilizar.

- Aprobación de una estructura orgánica. Es el nombramiento de responsables que se encargarán de la gestión y del control de la base de datos.

- Plan de trabajo detallado. Se trata de especificar las distintas fases con los plazos y medios que requieren cada una de ellas.

Concepción de la base de datos y selección del equipo.

- Concepción. El administrador de la base de datos junto con los usuarios se encargan de concretar la percepción del mundo real no sólo con respecto a los datos sino también a nivel procedimental creando un esquema conceptual. Este esquema es totalmente independiente de la máquina donde será instrumentado el sistema y del sistema de gestión de bases de datos utilizado, dependiendo sólo de las características del sistema actual.

- Especificación de las necesidades de equipo físico y lógico. Se determinan las exigencias en cuanto al dimensionamiento del equipo, es decir, cantidad de memoria principal y secundaria, capacidad de proceso, terminales, etc.

Diseño y carga.

- Fase de diseño. Cubre tanto el diseño lógico, como el diseño físico. El diseño lógico depende del modelo de datos de sistema de gestión de bases de datos elegido, jerárquico, red, relacional, objeto-relacional, u orientado a objetos. Se han de considerar aspectos de normalización, de conservación de semántica y de compromiso entre necesidades de usuarios y recursos existentes ya que puede haber muchos esquemas lógicos para un mismo esquema conceptual dado. El esquema físico se define a partir del esquema lógico, y pretende entre otras cosas mejorar los caminos de acceso a los datos para así optimizar el rendimiento de la base de datos.

- Carga y optimización. Consiste en cargar automática o manualmente los datos en la base de datos. Además se han de desarrollar (de forma paralela al diseño físico) los programas y procedimientos necesarios para instrumentar las reglas de ges-

ción que se definieron en la fase de concepción de la base de datos

Producción.

Es la fase de explotación del sistema de información basado en un sistema de gestión de bases de datos.

Una vez analizadas las fases de la vida de una base de datos, hemos de plantearnos qué queremos estimar, qué factores hemos de contemplar, que aspectos hemos de medir, y en qué momento deseamos estimar. A nivel general, nosotros intentamos contestar a esas preguntas marcándonos los

“Se han de considerar aspectos de normalización, de conservación de semántica y de compromiso entre necesidades de usuarios y recursos existentes.”

siguientes objetivos.

- Se desea estimar el esfuerzo de la puesta en marcha de la base de datos, es decir, desde el estudio de viabilidad hasta la carga de los datos.

- Los factores que se pueden contemplar son muy diversos y de distinta naturaleza, como el tipo de sistema de gestión de base de datos, los esquemas conceptuales, lógicos y físicos, los procedimientos que definen las reglas de gestión, los tipos de datos utilizados, la cantidad de datos a cargar, incluso si la carga se hacen manual o automáticamente. Se puede observar que son muy diversos los factores que pueden influir en las estimaciones, por lo tanto en los apartados siguientes, dependiendo del método de estimación utilizado, se podrán ver qué factores se tienen en cuenta para estimar, y de cuáles se prescinde.

- Para estimar es necesario medir, por

lo tanto hay que basar las estimaciones en mediciones de aspectos de otros sistemas ya existentes y del nuevo sistema. Los aspectos susceptibles de medición son aquellos subproductos que se obtienen en algunas etapas como son los esquemas conceptuales, lógicos y físicos.

- Las estimaciones de esfuerzo y coste deben ser realizadas al comienzo de la creación del sistema para poder ayudar a determinar una planificación adecuada y un presupuesto ajustado, que consiga dar beneficios a la empresa desarrolladora y que no sea prohibitivo para la empresa compradora. También es interesante realizar estimaciones a lo largo del desarrollo progresivamente para así ir teniendo datos cada vez más exactos. El problema que existe con las bases de datos es que hasta que no se han realizado las primeras etapas del ciclo de vida, no se dispone de ningún esquema que pueda ser medido. Al principio del ciclo de vida de las bases de datos, la única alternativa de estimación es aplicar la técnica del "Juicio Experto" para que en base a los requisitos iniciales y a la experiencia de los expertos, sea posible disponer de unas cantidades aproximadas del coste del sistema y de los recursos o esfuerzo necesario para su construcción. El siguiente momento dentro del ciclo de vida donde pueden ser realizadas estimaciones es en la concepción del sistema, puesto que está disponible un esquema, el conceptual, donde se pueden medir ciertos aspectos, y en función de esos aspectos medidos, se pueden establecer reglas que nos aproximen el esfuerzo y coste total del sistema. Posteriores refinamientos de las estimaciones se realizarán en la fase de diseño, donde ya tendremos el esquema lógico y el físico.

Como se puede observar, hay campo suficiente para definir un marco completo de estimación que monitoree la construcción de las bases de datos, obteniendo estimaciones paso a paso, para ir refinando las predicciones y así poder ir ajustando la planificación del proyecto acordeamente a las desviaciones producidas a medida que avanza la creación del proyecto. Por el momento, sería demasiado ambicioso por nuestra parte intentar abarcar tal esquema de estimación progresiva y por niveles, y nos limitaremos a proponer modelos para realizar estimaciones del coste y del esfuerzo de desarrollo a partir del esquema conceptual, que es el primero que se encuentra disponible.

■ 3 EL JUICIO EXPERTO PARA ESTIMAR AL PRINCIPIO DEL CICLO DE VIDA

Una explicación detallada de este método de estimación puede ser encontrada en Hughes (1996). Este método de estimación, como su nombre indica se basa en que una serie de personas consideradas 'expertas' predican el resultado de un aspecto del desarrollo del software, en este caso el coste o el esfuerzo. A pesar de parecer algo inseguro y de naturaleza subjetiva e inestructurada, la opinión experta es ampliamente utilizada en muchas ocasiones, ya que se confía más en la opinión de una persona con cierto bagaje en estas cuestiones que en los resultados de complicados sistemas matemáticos (Williams, 1995).

Para estimar en las primeras etapas del ciclo de vida de una base de datos, este método es el más adecuado, ya que aun no se dispone de ningún esquema representativo de la base de datos que pueda ayudar a realizar una estimación más objetiva, por lo tanto asumiremos que la opinión experta es usada para establecer una predicción inicial pero que se puede desviar más o menos en función de quien la realice. La exactitud también depende de la disponibilidad de información. A medida que el proyecto avanza, mayor es la información disponible y mayor será la seguridad de las estimaciones. Estas primeras estimaciones serán realizadas en la fase del ciclo de vida de la base de datos denominada Estudio Previo y Plan de Trabajo, concretamente en la subtarea conocida como Evaluación Previa de Medios y Costes, donde se ha de proporcionar una valoración (estimación) de los recursos necesarios para desarrollar la base de datos.

Para mejorar estas estimaciones se suelen utilizar unas técnicas llamadas Delphi y Dephi de banda ancha (Piattini, et al., 1996), que están basadas en el siguiente proceso:

- El coordinador proporciona a cada uno de los expertos una especificación del proyecto propuesto y un formulario donde éste expresará su opinión.
- Los expertos rellenarán el formulario sin ninguna interacción con el resto de expertos, aunque sí con el coordinador.
- El coordinador muestra a todos los expertos el valor medio de las opi-

niones para que éstos la comparen con la suya propia y puedan realizar una nueva estimación teniendo en cuenta la media.

■ Se repite el proceso hasta que se llegue a un consenso.

El método Delphi de banda ancha evita que los expertos estén aislados, fomentando el diálogo entre ellos y el intercambio de posturas con respecto a la estimación que cada uno realiza. De esta forma se consigue que detalles que algún experto no ha tenido en cuenta por error no influyan negativamente en la estimación.

Una persona es considerado un experto en estimación si cumple una o más de las siguientes características:

- Tiene trabajos prácticos en el área de aplicación que se quiere estimar.
- Tiene experiencia en supervisar o dirigir temas del área a estimar.
- Tiene experiencia en hacer estimaciones (aunque no sea en el área de aplicación en la que se desea estimar).

La principal conclusión es que el juicio experto no es una técnica demasiado fiable, aun-que sí adecuada como primera aproximación.

■ 4 ESTIMACIÓN POR ANALOGÍA PARA BASES DE DATOS

Las estimaciones que se pueden realizar en la fase del Estudio Previo y Plan de Trabajo, serán habitualmente de poca calidad, debido a la falta de información concreta de que se dispone en ese punto del ciclo de vida. Por lo tanto, resulta necesario realizar estimaciones más objetivas, basándonos sobre algún esquema o documento que ofrezca una visión lo más completa y concreta posible del sistema. A lo largo del ciclo de vida, el momento más temprano en el que disponemos de algún esquema representativo es en la Concepción del sistema, puesto que fruto de esta etapa surge un esquema entidad-interrelación, sobre el cual se pueden medir ciertos aspectos que pueden estar correlacionados con el esfuerzo de desarrollo total.

Una descripción completa de esta técnica de estimación puede ser encontrada en Catherwood (1998). La estimación por analogía consiste en encontrar semejanza, o parecido entre ciertas partes del modelo considerado.

Para poder aplicar esta técnica hemos de tener disponible una base de proyectos suficientemente amplia. Esta aproximación permite realizar estimaciones de tamaño y esfuerzo de sistemas, pero dicha estimación será orientativa, ya que nos basamos en las similitudes del nuevo sistema con otros ya existentes, y a menudo sucederá que tales similitudes no sean tan evidentes.

Esta técnica aplicada a esquemas entidad-interrelación tendría las siguientes fases:

- 1 Dividir el esquema en varias vistas independientes y de un tamaño relativamente pequeño, como pueden ser en subgrafos conexos de entidades, y asignar porcentajes de tamaño aproximados de cada parte con respecto del sistema global.
- 2 Comparar las vistas obtenidas con otras vistas de sistemas ya existentes. Para poder realizar este paso, es necesario tener disponibles las vistas de otros proyectos ya realizados, y con una estimación porcentual de cada una con respecto del sistema completo. Como parece evidente no será posible comparar el nuevo sistema completamente con otro ya existente, sino que compararemos ciertas partes del nuevo sistemas con partes de varios sistemas ya existentes. Es posible que tengamos otras partes que no estarán cubiertas por ningún sistema existente.

Todas las posibles combinaciones de vistas que podemos encontrar dependiendo de si existen otras parecidas o análogas en otros sistemas son las siguientes:

- a) **Vistas cubiertas por un sistema ya existente.** Esta vista será comparada con otra análoga de un sistema ya construido.
- b) **Vista cubierta de manera múltiple.** Será una zona del esquema entidad-interrelación que se pueda comparar con ciertas zonas o vistas de varios sistemas ya creados.
- c) **Vista no cubierta.** No existe ningún sistema que pueda ayudarnos a estimar esta zona por no tener ninguna similitud con el que estamos construyendo.

- 3 Determinar el esfuerzo estimado del nuevo sistema propuesto. La estimación será el resultado de sumar los esfuerzos estimados para cada una de las vistas del sistema.

ESTIMACIÓN DEL ESFUERZO

Suma del esfuerzo estimado de cada una de las vistas cubiertas por un sólo sistema existente
 +
 Suma del esfuerzo estimado de cada una de las vistas cubiertas por varios sistemas existentes
 +
 Suma del esfuerzo estimado de cada una de las vistas no cubiertas por ningún sistema existente

La cantidad de similitud o analogía entre varios sistemas se puede medir en términos de número de entidades análogas con respecto del esquema completo. El porcentaje de entidades análogas nos indicará también el porcentaje de esfuerzo (suponiendo que el tamaño del sistema medido en número de entidades está directamente correlacionado con el esfuerzo de desarrollo), que es el que consideraremos en la estimación, y que mediremos en personas-mes.

Veamos como podemos realizar esas estimaciones:

a) **Vista cubierta por otro sistema:** Se estima con la cantidad de personas-mes del sistema ya existente correspondiente al porcentaje de entidades análogas o similares.

b) **Vista cubierta por múltiples subsistemas:** Estimamos el esfuerzo realizando la media de los esfuerzos estimados por cada sistema existente.

c) **Vista no cubierta:** Esta estimación la hemos de realizar una vez que ya se han estimado el resto de vistas cubiertas por otro u otros sistemas. Si el porcentaje de entidades análogas a otros sistemas es el P%, el de no cubiertas es el Q% (P+Q=100) y Z es la cantidad de esfuerzo estimado de las entidades análogas a otros sistemas, el resultado de estimar el esfuerzo asociado a las entidades para las que no se ha encontrado similitud será (Z/P)*Q.

Consideremos el ejemplo de la Figura 2, donde el esquema entidad interrelación de una nueva base de datos puede ser cubierto al menos parcialmente por los esquemas entidad-interrelación de otras dos bases de datos que ya han sido construidas, y para los que se conoce su esfuerzo en personas-mes. Las estimaciones de cada vista son obtenidas de la siguiente forma:

⇒ Estimación de la zona cubierta (solamente) por A. 30% de A=15 P-M.

⇒ Estimación de la zona cubierta (solamente) por B. 35% de B =14 P-M.

⇒ Estimación de la zona múltiplemente cubierta.

10% de A 5 P-M.

15% de B 6 P-M

11 P-M / 2=5'5 P-M

⇒ Estimación de la zona no cubierta por ningún sistema.

El 75% del sistema total supone 15+14+5'5 = 34'5 P-M.

Entonces el 100% supone 46 P-M.

Por lo tanto, la parte no cubierta, el 25% supone 11'5 P-M.

⇒ Estimación del sistema completo. Es la suma de las estimaciones para las zonas no cubiertas y de las cubiertas:

34'5+11'5= 46 P-M.

De esta forma, ya tendríamos una estimación del esfuerzo de creación de la base de datos. La calidad de las estimaciones

dependerán del índice de correlación entre el porcentaje de entidades en el esquema entidad-interrelación y el porcentaje de esfuerzo consumido por esas entidades.

5 ESTIMACIÓN DEL TAMAÑO FUNCIONAL

En el campo de la estimación, las medidas del tamaño funcional han tenido mucha importancia a lo largo de la historia, desde la métrica Bang (DeMarco, 1982), con un campo de utilización limitada, hasta la métrica Puntos de Función (Albrecht, 1979), ampliamente utilizada, y apoyada por organizaciones como el International Function Points Group (IFPUG), y de la cual han aparecido muchas variantes para distintos paradigmas de desarrollo, como la técnica conocida como Full Function Points (Maya, et al., 1998; Oligny, et al., 1998; St-Pierre, et al., 1997), que fue especialmente diseñada para sistemas de tiempo real. El tamaño funcional es una medida del tamaño del sistema que percibe el cliente, siendo tal tamaño independiente de factores tecnológicos, aunque se utilicen en algunos casos para realizar ajustes y así poder encontrar correlación con factores como el esfuerzo de desarrollo. En las siguientes líneas se ofrece un repaso de la métrica Bang y Puntos de Función, y se busca una forma de adaptar estas técnicas al desarrollo de bases de datos.

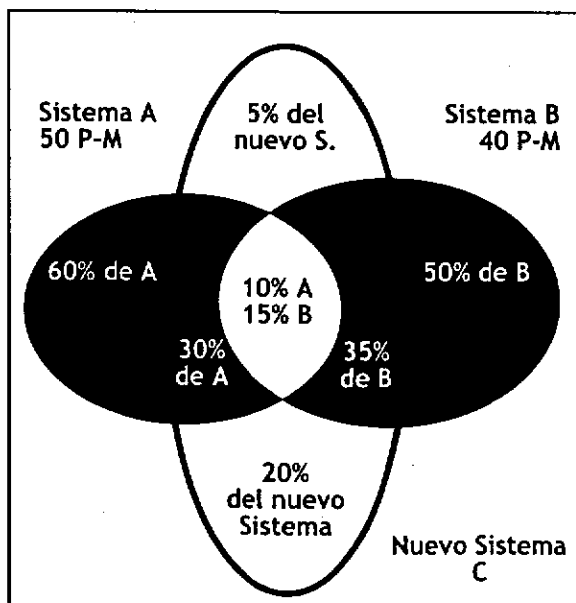


FIGURA 2 • Ejemplo de Particiones de un Sistema

5.1. Métrica Bang.

En DeMarco (1982), se afirma que el tamaño de la especificación de un sistema es una medida directa de la cantidad de funcionalidad útil del sistema. Para medir tal tamaño considera una medida conocida como Bang, y que se basa en los siguientes aspectos relacionados con las primitivas (componentes que no se pueden dividir en otros componentes subordinados) que se puede identificar en una especificación del sistema:

- **Primitiva funcional.** Es cada uno de los procesos indivisibles del diagrama de flujo de datos del modelo funcional.
- **Elemento de datos.** Es cada uno de los datos elementales (de tipo básico, no agregado) del diccionario de datos del modelo funcional.
- **Objetos.** Es un agrupamiento de items de datos almacenados y que están referidos a la misma entidad.
- **Interrelación.** Es la primitiva de interconexión entre datos.
- **Estado.** Es una primitiva del modelo de transición de estados.
- **Transición.** Es una segunda primitiva, de más alto nivel del modelo de transición de estados.

Para medir algunos de estos aspectos con un nivel de detalle más profundo son definidas las métricas primitivas siguientes:

- **FP.** El número de primitivas funcionales dentro del límite de la aplicación.
- **FPM.** El número de primitivas funcionales que están fuera del límite de la aplicación, pero que hay que modificar o tratar de cierta forma.
- **DE.** El número de datos elementales existentes dentro del límite de la aplicación.
- **DEI.** La cantidad de datos elementales de entrada.
- **DEO.** Son los datos elementales de salida.
- **DER.** Los datos elementales almacenados de forma automatizada.
- **OB.** Número de objetos en el modelo de datos.
- **RE.** El número de relaciones en el modelo de datos.
- **ST.** Cantidad de estados en el modelo de transición de estados.
- **TR.** Cantidad de transiciones en el modelo de transición de estados.
- **TCi.** Es el número de datos que están implicados con la primitiva funcional i , y que

no son subdivididos en primitivas.

- **REI.** El número de relaciones implicadas con el objeto i del modelo de datos.

Una primera aproximación para la construcción de la métrica Bang fue considerar una expresión que utilice todas estas medidas, y cada una multiplicada por un peso que se determinaría empíricamente:

$$Bang = W_{FP} FP + W_{DE} DE + \dots$$

El problema de esta aproximación es que la calidad de la métrica depende de la exactitud de los pesos aproximados mediante un análisis de regresión, y al ser demasiados valores, no es sencillo encontrar unos pesos que identifiquen correctamente la proporción de influencia de la variable asociada. Por lo tanto, la alternativa por la que se optaba era diferenciar varios tipos de sistemas, los intensivos en funcionalidad, los intensivos en datos y los mixtos. La forma de elegir el tipo de sistema era utilizando un ratio (RE/FP) y estableciendo unos intervalos para éste con los que se clasificaba cada sistema. De tal forma que la métrica Bang es adaptada dependiendo del tipo del sistema, considerando sólo aquellos aspectos que están relacionados con el perfil en que se habían considerado intensivos. En este trabajo nos centraremos en los sistemas intensivos en datos para después obtener similitudes con los sistemas en los que se desarrollan bases de datos. El principal aspecto que se considera para medir el tamaño funcional de un sistema intensivo en datos son los objetos que intervienen en la base de datos. Estos objetos eran ponderados aplicando unos pesos que están en función del número de relaciones que tengan tales objetos con el resto de objetos:

$$Bang_{Datos} = \sum_i^{N^{\circ} \text{Objetos}} Peso_i$$

Una vez que se dispone de esta medida del tamaño funcional, se pueden realizar estimaciones del esfuerzo de desarrollo mediante un análisis de correlación.

Esta forma de medir el tamaño de los sistemas puede ser adoptada directamente por el paradigma de las bases de datos, puesto que se consideran unos aspectos que

están disponibles en un esquema entidad-interrelación, que es la parte más representativa de la base de datos en las primeras etapas del ciclo de vida. Por lo tanto, una alternativa más para poder realizar estimaciones del esfuerzo de desarrollo de bases de datos es aplicar directamente la métrica Bang en su versión para sistemas intensivos en datos, sin considerar el resto de aspectos funcionales, sólo considerando todas las entidades, y dándole como peso a cada una el número de interrelaciones en las que participe.

5.2. Métrica Puntos de Función.

El método de estimación de costes y de esfuerzo basado en puntos de función es muy conocido bajo las siglas FPA (Function Point Analysis). Pretende medir el tamaño del sistema con respecto a la funcionalidad que percibe el cliente, y no con respecto a otros aspectos más tangibles como podrían ser las líneas de código fuente. Se utiliza la medida Punto de Función, que es obtenida tras una serie de etapas y de ajustes. El tamaño funcional de los sistemas medido en puntos de función está correlacionado con el esfuerzo de desarrollo, de ahí que sea un método muy útil de estimación, ya que los aspectos que se miden se pueden determinar en las primeras etapas del ciclo de vida del software, y una vez que se dispone del tamaño funcional se puede conseguir una aproximación del esfuerzo necesario para construir el sistema.

A pesar de que los puntos de función han sido y son muy utilizados para realizar predicciones del coste y esfuerzo de desarrollo, existen una fuerte discusión sobre la rigurosidad de tal método, así como de su utilidad práctica (Fernández, 1999).

Los aspectos que son medidos en esta técnica son los siguientes:

- El número de entradas. Un flujo de información de entrada.
- El número de salidas. Son flujos de información de salida.
- Los ficheros lógicos internos. Son datos que gestiona la aplicación.
- El número de ficheros de interfaz externos. Son datos que la aplicación usa pero que no gestiona.
- Las consultas externas. Son transacciones en las que se produce una entrada y una salida a continuación.

El proceso consiste en medir los aspectos considerados antes, y ponderarlos dependiendo de una complejidad subjetiva (alta, baja o media) que se le asigna a cada aspecto. Una vez que se realiza el recuento, la suma que se obtiene se conoce como Puntos de Función Sin Ajustar. Esta medida es ajustada mediante la consideración de unos factores tecnológicos como la comunicación de datos, el rendimiento, la frecuencia de transacciones, la dificultad de procesos, la existencia o no de funciones distribuidas, etc., dándole un valor entero de 0 a 5 dependiendo de la importancia o la dificultad que aporte cada aspecto. Todos estos factores, podrán variar en un (35% el valor de los puntos de función sin ajustar, y el resultado da lugar finalmente a los puntos de función.

Esta técnica de estimación puede ser redirigida hacia el campo de las bases de datos, puesto que se pueden considerar sólo los aspectos referentes a datos, como los ficheros internos lógicos y los ficheros de interfaz externos, dejando de lado todo lo referente a funciones, como las entradas, salidas y consultas. En las siguientes líneas se muestra una adaptación del método de puntos de función para medir el tamaño funcional al ámbito de las bases de datos.

Para medir el tamaño de una base de datos y utilizar ese dato para pronosticar el esfuerzo de desarrollo necesitamos medir el tamaño del esquema entidad-interrelación asociado a la base de datos. De entre todos los aspectos medidos en la técnica de puntos de función, sólo los ficheros lógicos internos y los ficheros externos de interfaz son útiles para medir el tamaño de un esquema entidad-interrelación (Muller, 1999).

Un fichero será considerado como un fichero lógico interno si los datos son persistentes y si se crean y manipulan desde dentro de los límites

de la aplicación. En cambio, un fichero de interfaz externo será aquel que es manejado pero no mantenido, es decir, ni creado ni modificado dentro de los límites del sistema. La mayoría de sistemas tienen muchas pequeñas tablas que son mantenidas por un administrador como listas de ciudades o estados, códigos de tipos de datos, información de seguridad, etc. Esas tablas suelen considerarse ficheros externos de interfaz.

Siguiendo las pautas marcadas por la técnica de puntos de función, la primera etapa para medir el tamaño funcional del sistema es contar el número de ficheros lógicos internos y el de ficheros de interfaz externos. El número de esos ficheros con los que cuenta el sistema podría ser una medida del tamaño funcional, pero se suele refinar esa medida dando mayor valor a aquellos ficheros que tengan más complejidad, y menos a los que tengan una complejidad menor. La complejidad de cada uno de los ficheros, tanto internos como externos de interfaz, se calcula en función de los siguientes factores:

• **RET:** Son subgrupos de datos elementales reconocibles por el usuario que se encuentran contenidos en los ficheros lógicos internos o en los ficheros externos

de interfaz.

• **DET:** Son atributos reconocibles por el usuario o campos atómicos, no recursivos que pueden ser claves externas en una entidad.

Una vez contados el número de RETs y el de DETs para cada fichero lógico interno y para cada fichero de interfaz externo se calcula la complejidad de cada uno

mediante la tabla 2, que es la utilizada en el método de análisis de puntos de función para calcular la complejidad de este tipo de ficheros:

Una vez que hemos determinado la complejidad de cada uno de los ficheros lógicos internos y externos de interfaz, se pueden obtener los llamados puntos de función sin ajustar como resultado de sumar por cada fichero un peso dependiente de su complejidad que se consigue a través de la tabla 3, y que son los datos propuestos para la técnica de puntos de función para medir el tamaño funcional de los sistemas de información.

El siguiente paso de acuerdo con FPA sería el cálculo de los puntos de función, utilizando la medida de puntos de función sin ajustar y un factor de ajuste obtenido como resultado de analizar unos factores relacionados con las características generales del siste-

"Un fichero de interfaz externo será aquel que es manejado pero no mantenido, es decir, ni creado ni modificado dentro de los límites del sistema"

RETs/DETs	1-19	20-50	51-
1	Baja	Baja	Media
2-5	Baja	Media	Alta
6-	Media	Alta	Alta

TABLA 2 • Complejidad de los ficheros

Peso/Complejidad	Baja	Media	Alta
Fichero Lógico Interno	7	10	15
Fichero Externo de Interfaz	5	7	10

TABLA 3 • Ponderación de los ficheros dependiendo de su Complejidad

ma. En Muller, (1999) se indica que no es demasiado aconsejable realizar este paso debido a la degradación teórica que sufre el cálculo de puntos de función frente al cálculo de puntos de función sin ajustar al

incluir una escala ordinal (de 0 a 5) para medir tales características generales del sistema como la comunicación de datos, el ratio de transacciones, la reusabilidad, etc. Por lo tanto, optamos por recomendar la

finalización del cálculo de puntos de función en este punto utilizando como medida última la de los puntos de función sin ajustar. ■

CONCLUSIONES

En este artículo, se ha justificado la necesidad de la existencia de métodos para estimar el coste y el esfuerzo de la construcción y puesta en marcha de bases de datos, considerando esta fase como una de las principales en el proceso de desarrollo de sistemas de información automatizados. Como se ha podido observar, las estimaciones se realizan partiendo de datos recogidos sobre proyectos anteriores, pero considerando que han de ser ajustados de acuerdo al entorno concreto para el que se realizan las estimaciones. Se ha mostrado que uno de los métodos de estimación que mayor repercusión ha tenido a lo largo de la historia es el de Puntos de Función, pero actualmente existen discrepancias a cerca de su calidad, y se desconfía de su validez. El problema de las estimaciones es irresoluble y sólo cabe encontrar métodos que nos ayuden dándonos datos aproximados que nos orienten cuantitativamente sobre ciertos aspectos del desarrollo, y que justifiquen las decisiones tomadas por los expertos.

BIBLIOGRAFÍA

- Abdel-Hamid, T. (1993). *Adapting, Correcting, and Perfecting Software Estimates: A Maintenance Metaphor*. IEEE Computer Society. March 1993. Pp. 20-29.
- Albrecht, A.J. (1979). *Measuring Application Development Productivity*. Proceedings of the Joint SHARE, GUIDE and IBM Application Development Symposium, October 1979. Pp. 83-92.
- Gatherwood, B. AND Gupta, M. (1998) *Life-Cycle Estimation for Object Oriented Systems. Prediction by Analogy*. Proceeding of the FESMA98 Conference, 6-8 Mayo 1998.
- DeMarco, T. (1982). *Controlling Software Projects. Management, Measurement & Estimation*. Englewood Cliffs, NJ.: Prentice Hall.
- De Miguel, A. y Piattini, M. (1993). *Concepción y diseño de bases de datos. Del modelo ER al modelo relacional*. Madrid, Ra-Ma.
- Ebrahimi, N.B. (1999). *How to improve the Calibration of Cost Models*. IEEE Transactions on Software Engineering. Vol 25, nº 1, January/February 1999, pp. 136-140.
- Fernández, L. (1999). *Una revisión breve de la medición de software*. NOVATI-CA, Nº 137, 1999. Pp. 20-24.
- Graham, I. (1995). *Migrating to Object Technology*. Addison-Wesley. Wokingham, England.
- Hughes, R. (1996). *Expert judgement as an estimating method*. Information and Software Technology. 1996. Elsevier. 38. Pps. 67-75.
- Jones, C. (1996). *Applied Software Measurement*. 2nd Ed. McGraw Hill.
- Maya, M., Abran, A., Oigny, S., St-Pierre, D. And Desharnais, J-M. (1998). *Measuring the functional size of real-time software*. Proceeding of the 9th European Software Control and Metrics Conference, Rome Italy, May 27-29.
- Muller, R. J. (1999) *Database Design for Smarties - Using UML for Data Modeling*. Morgan Kaufmann Publishers, Inc.
- Oigny, S., Abran, A., Desharnais, J-M. And Morris, P. (1998). *Functional Size of Real-Time Software: Overview of Field Tests*. Proceeding of the 13th International Forum on COCOMO and Software Cost Modeling. Los Angeles, CA, October 6-8, 1998.
- Piattini, M., Calvo-Manzano, J., Cervera, J. y Fernández, L. (1996). *Análisis y diseño detallado de Aplicaciones Informáticas de Gestión*. Madrid, Ra-Ma.
- St-Pierre, D., Maya, M., Abran, A., Desharnais, J-M., And Bourque, P. (1997). *Full Function Points: Function Points Extension for Real-Time Software - Counting Practices Manual*. Université du Québec à Montréal. Montréal, Technical Report nº 1997-04. September 1997.
- Williams, J. (1995). *What Every Software Manager MUST KNOW TO SUCCEED with Object Technology*. SIGS Books, New York.
- Zuse, H. (1998). *A Framework of Software Measurement*. Walter de Gruyter.