

# V Jornadas

8, 9 y 10 de noviembre  
Valladolid

**di**  
Departamento de  
Informática

EDITORES:  
Carlos Delgado  
Esperanza Marcos  
José M. Marqués

ACTAS DE LAS V JORNADAS DE INGENIERÍA DEL SOFTWARE  
Y BASES DE DATOS (JISBD2000)

EDITORES:

Carlos Delgado  
Esperanza Marcos  
Jose M. Marqués

ORGANIZADAS POR:

Departamento de Informática  
Universidad de Valladolid

ENTIDADES COLABORADORAS:

Universidad de Valladolid. Vicerrectorado de Investigación  
Microsoft  
Uno-e  
Prentice-Hall  
Tecsidel  
Ayuntamiento de Valladolid

© Los autores

Primera edición, 2000

I.S.B.N.: 84-8448-065-8

Deposito Legal: VA-800/2000

Imprime: Gráficas Andrés Martín S.L.

## COMITÉ DE PROGRAMA DE BASES DE DATOS

### PRESIDENTA

Esperanza Marcos (Universidad Rey Juan Carlos)

### MIEMBROS

José Francisco Aldana	(Universidad de Málaga)
M <sup>a</sup> José Aramburu	(Universidad Jaime I)
Pedro Blesa	(Universidad Politécnica de Valencia)
Matilde Celma	(Universidad Politécnica de Valencia)
João Correia Lopes	(Universidade do Porto)
Dolors Costal	(Universidad Politécnica de Cataluña)
Gabriel David	(Universidade do Porto)
Oscar Díaz	(Universidad del País Vasco)
Jesús García Molina	(Universidad de Murcia)
Alfredo Gofii	(Universidad del País Vasco)
Arantza Illaramendi	(Universidad del País Vasco)
Paloma Martínez	(Universidad Carlos III de Madrid)
Eduardo Mena	(Universidad de Zaragoza)
Pablo de la Fuente	(Universidad de Valladolid)
Joaquim Nunes Aparício	(Universidade Nova de Lisboa)
Rui Oliveira	(Universidade do Minho)
Mario Piattini	(Universidad Carlos III de Madrid)
Antonio Polo	(Universidad de Extremadura)
Nieves R. Brisboa	(Universidad de la Coruña)
Pilar Rodriguez	(Universidad Autónoma de Madrid)
Félix Saltor	(Universidad Politécnica de Cataluña)
José Samos	(Universidad de Granada)
Pedro Sousa	(Universidad Técnica de Lisboa)
Ernest Teniente	(Universidad Politécnica de Cataluña)
Miguel Toro	(Universidad de Sevilla)
Toni Urfí	(Universidad Politécnica de Cataluña)

### COLABORADORES EN EL PROCESO DE REVISIÓN

Albert Abello	Rafael Berlanga
Coral Calero	Gabriel David
Xavier Franch	Francisco José Galán
Jorge Enrique Martín	Francisco José Galán
Antonio Luis Sousa	

## COMITÉ DE PROGRAMA DE INGENIERÍA DEL SOFTWARE

### PRESIDENTE

Carlos Delgado Kloos (Universidad Carlos III de Madrid)

### MIEMBROS

Idoia Alarcón	(Universidad Autónoma de Madrid)
Pere Botella	(Universitat Politècnica de Catalunya)
Buenaventura Clares	(Universidad de Granada)
José Luis Fiadeiro	(Universidade de Lisboa)
Xavier Franch	(Universitat Politècnica de Catalunya)
Lidia Fuentes	(Universidad de Málaga)
Juan Garbajosa	(Universidad Politécnica de Madrid)
Juan Hernández	(Universidad de Extremadura)
Natalia Juristo	(Universidad Politécnica de Madrid)
José Manuel Marqués	(Universidad de Valladolid)
Ana Moreira	(Universidade Nova de Lisboa)
Juan José Moreno	(Universidad Politécnica de Madrid)
José Nuno Fonseca de Oliveira	(Universidade do Minho)
João Pascoal Faria	(Universidade do Porto)
Oscar Pastor	(Universitat Politècnica de Valencia)
Pedro Pastor	(Universidad de Alicante)
Mario Piattini	(Universidad de Castilla-La Mancha)
Isidro Ramos	(Universitat Politècnica de Valencia)
António Rito Silva	(Universidade Técnica de Lisboa)
Miguel Toro	(Universidad de Sevilla)
Ambrosio Toval	(Universidad de Murcia)
José María Troya	(Universidad de Málaga)
Javier Tuya	(Universidad de Oviedo)
Ángel Velázquez	(Universidad Rey Juan Carlos)
Juan Carlos Yelmo	(Universidad Politécnica de Madrid)

## COLABORADORES EN EL PROCESO DE REVISIÓN

José Carlos Bacelar  
José Bernardo Barros  
Orlando Belo  
José Hilario Canos  
Ana Cavalcanti  
Dolors Costal Costa  
Oscar Dieste  
Amador Durán  
José Luis Fernández  
Xavier Ferré  
Marcela Genero Bocco  
Rui Gustavo Crespo  
Ricardo Imbert  
Miguel A Laguna  
Esperanza Manso  
Manuel Mejías  
Begoña Moros  
Joaquín Nicolás  
Camilo Ocampo Goujon  
Macario Polo Usaola  
Francisco Ruiz González  
José A Troyano  
Sira Vegas Hernández

## COMITÉ ORGANIZADOR

### PRESIDENTE

José Manuel Marqués (Universidad de Valladolid)

### SECRETARÍA

Pablo de la Fuente (Universidad de Valladolid)

### MIEMBROS

Manuel Barrio	(Universidad de Valladolid)
Valentín Cardeñoso	(Universidad de Valladolid)
Carlos E. Cuesta	(Universidad de Valladolid)
Carmen Hernández	(Universidad de Valladolid)
Miguel A. Laguna	(Universidad de Valladolid)
Esperanza Manso	(Universidad de Valladolid)
Mercedes Martínez	(Universidad de Valladolid)
Félix Prieto	(Universidad de Valladolid)
Belarmino Pulido	(Universidad de Valladolid)
Juan J. Rodríguez	(Universidad de Valladolid)
Pilar Romay	(Universidad de Valladolid)
Jesús Vegas	(Universidad de Valladolid)
Diego R. Llanos	(Universidad de Valladolid)
Juan Hernández	(Universidad de Extremadura)
Mario Piattini	(Universidad de Castilla-La Mancha)
Ana Moreira	(Universidade Nova de Lisboa)

## PRÓLOGO

JISBD2000 reúne dos eventos ya consolidados, las Jornadas de Ingeniería del Software y las Jornadas de Bases de Datos que, en los últimos años, son referencia obligada para todos aquellos investigadores y profesionales interesados en conocer los últimos avances y tendencias en estas áreas. La celebración conjunta de ambos encuentros permitirá afianzar la experiencia iniciada, el pasado año en Cáceres, con tan buenos resultados científicos como organizativos.

En el presente volumen se recogen los trabajos presentados en las V Jornadas de Ingeniería del Software y V Jornadas de Bases de Datos, JISBD2000, celebrados en Valladolid los días 8, 9 y 10 de noviembre de 2000.

JISBD2000 se ha desarrollado en ocho sesiones técnicas, cinco talleres y cinco conferencias invitadas. A estas actividades se ha unido la celebración de las Primeras Jornadas de Bibliotecas Digitales y la Jornada de Seguimiento de proyectos CICYT-TIC-INFO.

Para participar en las sesiones técnicas se recibieron un total de 47 trabajos en Ingeniería del Software y 18 trabajos en Bases de Datos. El esfuerzo de los miembros de los respectivos comités de programa y de colaboradores externos, permitió realizar el proceso de revisión de forma satisfactoria. Nuestro agradecimiento a quienes enviaron sus trabajos y participaron en este proceso.

Cada artículo se revisó por tres evaluadores, quienes tras una ardua labor, dado el número y la calidad de las contribuciones recibidas, seleccionaron 17 trabajos en el área de Ingeniería del Software y 12 en el área de Bases de Datos. Adicionalmente, en el área de Ingeniería del Software, se aceptaron 6 trabajos para su presentación en la modalidad de artículos cortos y 6 en la de poster.

La realización de talleres se incorpora como novedad en esta edición de las JISBD. Su objetivo es proporcionar un foro de en el que un grupo de investigadores y profesionales intercambien opiniones, ideas o resultados sobre temas específicos. Los talleres programados han sido los siguientes: Aplicación de la Ingeniería del Software y las Bases de Datos a los Sistemas de Información Geográfica; Ingeniería del Software Basada en Componentes Distribuidos; Medición, experimentación y calidad en Ingeniería del Software; Bases de Datos Orientadas a Objetos y Lenguajes de especificación en el desarrollo del software.

JISBD2000 es el resultado del trabajo voluntario y desinteresado de muchas personas. Mi agradecimiento a los miembros de los Comités de Programa por su esfuerzo y dedicación, en especial a sus presidentes, Dr. Carlos Delgado y Dra. Esperanza Marcos, a los conferenciantes invitados, a los autores que enviaron sus trabajos y a todas las organizaciones que con su ayuda económica han hecho posible la realización de estas jornadas.

Por último, mi mas sincero reconocimiento a todos los miembros del Comité Organizador, por su entrega y entusiasmo a lo largo de todo el proceso de organización estas jornadas.

*José Manuel Márquez Corral*  
Presidente del Comité Organizador

# ÍNDICE

## Conferencias Invitadas

The Emerging Consensus on the Software Engineering Body of Knowledge <i>Pierre Bourque, Robert Dupuis, and Alain Abran</i>	3
WideAreaComputation <i>Luca Cardelli</i>	5
All Together Now. Towards Integrating the World's Information Systems <i>Klaus R. Dittrich &amp; Dirk Jonscher</i>	7
SoftwareEngineeringasaModellingProcess <i>Manfred Broy</i>	9
WebOntEx: Extracting Ontologies from Web Pages <i>Ramez Elmasi</i>	11

## Sesiones Técnicas de Ingeniería del Software

### ISP POSTER

Propuestas de Reutilización de Requisitos <i>Amador Durán Toro, Francisco José García Peñalvo, Miguel Angel Laguna Serrano</i>	17
Semantic Structure + Hypermedia = Cognitive Hypermedia <i>Lina García Cabrera, José Parets Llorca</i>	19
Verificación Formal del Sistema de Control de Acceso a Aplicaciones Corporativas de la F.L.C <i>Esther Sudrez-Calvo, Javier Tuyá</i>	21
Identificando Patrones de Diseño para Reuso de Código Legado Escrito en Lenguaje C++ <i>René Santaolaya Salgado, Máximo López Sánchez, Bárbaro J. Ferro Castro, Olivia G. Fragoso Diaz</i>	23

### ARTÍCULOS CORTOS

Primeros Pasos hacia un Compilador de UML/OCL <i>José Miguel Cañete Valdeón, Francisco José Galán Morillo, Miguel Toro Bonilla</i>	27
Aproximación metodológica al desarrollo de sistemas para el tratamiento de bibliotecas digitales <i>M.J. Escalona, M. Mejías, J. Torres</i>	33
Principios Básicos de Usabilidad para Ingenieros Software <i>Xavier Ferré Grau</i>	39
Cuestionario para la identificación de Riesgos en Proyectos de Mantenimiento <i>Macario Polo Usola, Mario Piattini Velthuis</i>	47
Verificación Modular de Sistemas Reactivos con Abstracciones del Entorno <i>Claudio de la Riva, Javier Tuyá, Jose Ramon de Diego</i>	53
El Proceso del Desarrollo en las aplicaciones Web: Estudio de la aplicabilidad de las metodologías de desarrollo existentes <i>Paloma Cáceres, Esperanza Marcos</i>	59

## IS1 INGENIERÍA DE REQUISITOS

- OCA: Un entorno de animación concurrente para la validación de modelos conceptuales ..... 67  
*Pedro Sánchez Palma Patricio Letelier Torres Isidro Ramos Salavert*
- Una Aproximación Ascendente al Modelado de Flujos de Trabajo ..... 79  
*José H. Canós, Juan Sánchez y M<sup>a</sup> Carmen Penadés*
- Generación Automática de Casos de Uso para Desarrollo de Software Basado en Reutilización ..... 89  
*Oscar López, Miguel Angel Laguna, José Manuel Marqués*
- De los procesos del negocio a los casos de uso ..... 103  
*Jesús García Molina, M. José Ortín Ibáñez, Begoña Moros Valle, Joaquín Nicolás Ros, Ambrosio Toval Álvarez*
- Marco de trabajo para la Evaluación de las Consecuencias de los Requisitos ..... 117  
*Oscar Dieste, Andrés Silva*

## IS2 MANTENIMIENTO, EVOLUCIÓN Y CALIDAD

- Validación de medidas para evaluar la mantenibilidad de programas SQL ..... 131  
*Antonio Martínez Hernández Mario Piattini Velshuis*
- Sobre la Búsqueda y Emparejamiento de Componentes COTS con Múltiples Interfaces ..... 141  
*Luis Iribarne y Antonio Vallecillo*
- Mejoras en la toma de decisiones en proyectos de software.  
 Aplicaciones de técnicas de lógica borrosa ..... 153  
*Javier Aroba Isabel Ramos José Riquelme*
- Mecanos y Análisis de Conceptos Formales como soporte para la construcción de Frameworks ..... 163  
*Félix Prieto, Yania Crespo, José Manuel Marqués, Miguel Angel Laguna*
- Equilibrio entre eficiencia y adaptabilidad usando separación de aspectos en la fase de diseño ..... 177  
*J.L. Herrero, M. Sánchez, F. Sánchez, M. Toro*

## IS3 ARQUITECTURAS SOFTWARE

- Una aproximación al desarrollo de software basada en prototipado arquitectónico ..... 191  
*Joaquín Peña, Rafael Corchuelo, Antonio Ruiz, Francisco Ferrer y Miguel Toro*
- Arquitectura de Software Dinámica basada en Reflexión ..... 203  
*Carlos E. Cuesta Quintero, Pablo de la Fuente Redondo, Manuel Barrio Solórzano, Encarnación Beato Gutiérrez*

## IS4 MÉTODOS DE DESARROLLO SOFTWARE

- Towards Automating the Iterative Rapid Prototyping Process with the SLAM System ..... 217  
*Ángel Herranz-Nieva, Juan José Moreno-Navarro*
- Una Metodología de Desarrollo Orientada a Objetos para Sistemas Empotrados de Tiempo Real .. 229  
*J.M. Álvarez, M. Díaz, L. Llopis, E. Pimentel, J.M. Troya*
- FELAS: Entorno de Desarrollo de Interfaces de Usuario en C++ ..... 241  
*María Dolores Lozano, Pascual González, Isidro Ramos*

Modelando aspectos de navegación y presentación en aplicaciones hipermediales .....	253
<i>C. Cachero, J. Gómez y O. Pastor</i>	
Rigorously Transforming UML Class Diagrams .....	265
<i>J.L. Fernández Alemán, A. Toval Alvarez, J.R. Hoyos Barceló</i>	

## Sesiones Técnicas de Bases de Datos

### BD1 BASE DE DATOS 1

Revisión de la Capacidad de Información en Diferentes Modelos de Datos .....	283
<i>Cecilia Delgado, José Samos</i>	
Un Procedimiento de Optimización Semántica de Programas Datalog .....	295
<i>Jose R. Parama, Nieves R. Brisaboa, Miguel R. Penabad y Hector J. Hernandez</i>	
Describing BLOOM99 with regard to UML Semantics .....	307
<i>Alberto Abello y M. Elena Rodriguez</i>	
Databases and Natural Language Interfaces .....	321
<i>Porfirio P. Filipe &amp; Nuno J. Mamede</i>	
Comportamiento del árbol Q como método de acceso a datos multidimensionales de amplio espectro .....	333
<i>Elena Jurado y Manuel Barrera</i>	

### BD2 BASE DE DATOS 2

Diseño dirigido por restricciones. Una metodología de desarrollo de sistemas de bases de datos relacionales basada en el modelo Entidad-Relación .....	347
<i>Maitilde Celma, Juan Carlos Casamayor</i>	
A Measurement Approach for Conceptual Database Design .....	357
<i>Marcela Genero y Mario Piattini</i>	

### BD3 BASE DE DATOS 3

Métricas para la calidad de bases de datos activas .....	371
<i>Coral Calero, Mario Piattini, Óscar Díaz</i>	
Inclusión de Vistas en ODMG .....	383
<i>Ginés García Mateos, Jesús Joaquín García Molina</i>	
Evaluación de Políticas de Reemplazo de Página para Consultas de Pares Cercanos Utilizando R-trees .....	397
<i>Antonio Corral, José Samos</i>	
Una arquitectura multi-capa para la recuperación y presentación de información histórica de una base de documentos .....	409
<i>Carlos Granell, Juan Manuel Pérez, María José Aramburu y Rafael Berlanga</i>	
Optimización para particiones en un RDBMS .....	421
<i>Diego Sevilla Ruiz, Jesús García Molina</i>	

# MÉTRICAS PARA LA CALIDAD DE BASES DE DATOS ACTIVAS

Coral Calero<sup>1</sup>, Mario Piattini<sup>1</sup>, Óscar Díaz<sup>2</sup>

<sup>1</sup>Grupo ALARCOS

Universidad de Castilla-La Mancha

Ronda Calatrava, 5 13071 Ciudad Real

e-mail: {ccalero, mpiattin}@inf-cr.uclm.es

<sup>2</sup>Grupo EKIN

Universidad del País Vasco/ Euskal Herriko Unibertsitatea

Pº Manuel de Lardizabal, 1 20.080 San Sebastián

e-mail: jipdigao@si.ehu.es

## Resumen

Los datos suponen una parte fundamental de los sistemas de información. Conseguir controlar la complejidad aportada por ellos resulta, por tanto, imprescindible. Una forma de realizar este control es mediante la utilización de métricas específicas para bases de datos. Este artículo propone distintas métricas para medir la complejidad intra-regla de los disparadores SQL para bases de datos activas. Asimismo se presentan dos experimentos, llevados a cabo con el fin de validar las métricas propuestas.

**Palabras clave:** bases de datos activas, calidad, métricas, GQM, validación empírica

## Introducción

Las bases de datos se han constituido como el factor fundamental dentro de los sistemas de información. Por lo tanto, si queremos obtener sistemas de calidad, deberemos asegurar la calidad de las bases de datos. Para ello, las métricas son una

herramienta primordial. Las métricas del software son un buen medio para entender, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento (Briand et al., 1996), así como un medio para profesionales e investigadores puedan tomar las mejores decisiones (Pfleeger, 1997). En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar procesos y productos (Fenton y Pfleeger, 1997).

Este trabajo acomete la elaboración y validación de métricas para las bases de datos activas (Paton y Díaz, 1999). Estos sistemas incorporan la noción de reglas evento-condición-acción (reglas ECA), donde el evento describe un acontecimiento ante el cual la regla es capaz de responder, la condición se encarga de examinar el contexto en el que el evento ha tenido lugar y la acción describe la tarea que la regla realizará si el evento tiene lugar y la condición se cumple. La parte del evento puede ser un evento primitivo (por ejemplo, una operación en la base de datos), o un evento compuesto (combinación de eventos primitivos o eventos compuestos).

La posibilidad de trabajar con reglas ECA está incorporada ya, aunque de manera limitada, en un gran número de productos comerciales (Oracle, Sybase, DB2, SQL-Server, Informix) ya que estas reglas son consideradas altamente útiles en gran cantidad de aplicaciones (Widom y Ceri, 1996). Sin embargo, y a pesar de su utilidad, algunos informes recientes señalan inconvenientes asociados a las reglas ECA, relacionados, sobre todo, con el mantenimiento de las mismas (Kotz, 1994). Por tanto, en las bases de datos activas es especialmente importante poder controlar la mantenibilidad.

La mantenibilidad es uno de los factores que influyen en la calidad de un producto software según la ISO 9126. Los otros factores influyentes son funcionalidad, fiabilidad, usabilidad, eficiencia y portabilidad. La mantenibilidad, puede ser, a su vez, descompuesta en otras subcaracterísticas, como son: 'analizabilidad', 'cambiabilidad', estabilidad, facilidad de prueba y conformidad. Tanto la 'analizabilidad' como la 'cambiabilidad' y la facilidad de prueba dependen de la complejidad (Li y Chen, 1987): La complejidad puede ser dividida en computacional, psicológica y representacional (Henderson-Sellers, 1996). Además, este autor considera que la complejidad psicológica incluye la complejidad del problema, los factores humanos de tipo cognitivo y la complejidad del producto. Nuestro trabajo se centra precisamente en elaborar métricas para evaluar la complejidad de reglas ECA. Las métricas de producto pueden ser divididas en intra-modulares e inter-modulares (Kitchenham y Linkman, 1990). De esta forma, para las bases de datos activas podremos considerar tanto la complejidad intra-regla (cuando las reglas son medidas en sí mismas) como la inter-regla (cuando se mide la interacción entre reglas).

En este trabajo presentamos métricas intra-reglas. En Díaz et al. (2000) se abordan las métricas inter-regla.

La exposición se realiza según el siguiente guión. Primeramente, se proponen las métricas intra-reglas obtenidas para bases de datos activas mediante el método GQM

(Goal-Question-Metric). El apartado tercero muestra dos validaciones empíricas diferentes realizadas con las métricas. Por último, la sección cuarta presentará las conclusiones y los futuros trabajos a desarrollar.

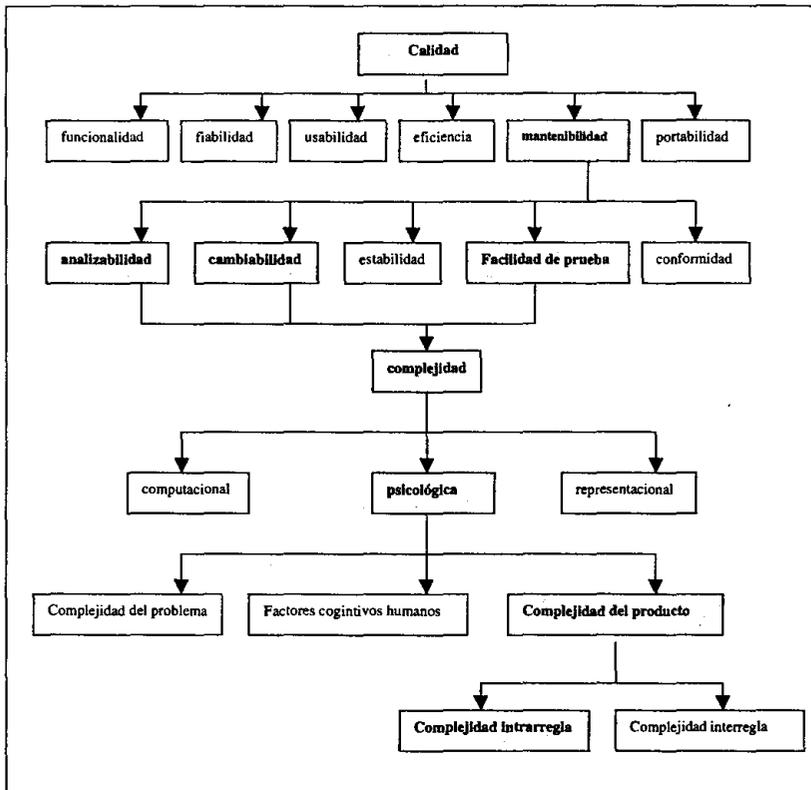


Fig. 1. Relación entre las métricas intra-regla para la complejidad del producto y la calidad del software

### Aproximación GQM

La aproximación GQM (Goal-Question-Metric) se basa en el hecho de que cualquier métrica puede ser definida mediante un esquema arriba-abajo. El resultado de aplicar la aproximación GQM es un modelo en tres niveles: el nivel conceptual en el que se definen los objetivos (goal), el nivel operacional en el que se definen las preguntas (question) y el nivel cuantitativo en el que se definen las métricas (metric). De esta forma, el objetivo se define mediante una serie de preguntas y cada pregunta se redefine a

través de unas métricas. En la tabla 1 se presenta la aproximación GQM para nuestro caso y, por tanto, las métricas propuestas:

Objetivo ( <i>goal</i> )	Pregunta ( <i>question</i> )	Métrica ( <i>metric</i> )
Propósito: <i>Mejorar el</i> Asunto: <i>mantenimiento</i> Objeto: <i>de las reglas ECA</i> Punto de vista: <i>desde el punto de vista del diseñador de la base de datos</i>	¿Cómo influye la complejidad de la condición en la mantenibilidad de la regla ECA?  ¿Cómo influye la complejidad de la acción en la mantenibilidad de la regla ECA?	NO. Número de Operandos que componen la condición NS. Número de Sentencias SQL incluidas en la acción

Table 1. Métricas obtenidas por la aproximación GQM

Así pues, tenemos como resultado de la aproximación GQM dos métricas intra-regla que nos permiten medir la calidad de las bases de datos activas. Estas dos métricas cubren dos de las tres partes que componen una regla ECA: la condición y la acción. No se ha considerado la posibilidad de definir métricas para la parte del evento ya que la mayoría de los productos comerciales se limitan a soportar eventos simples. Para la condición se propone como métrica el Número de Operandos que componen la condición del disparador (NO). Y para la parte acción, proponemos el Número de Sentencias SQL incluidas en la acción (NS<sup>1</sup>).

En la figura 2, presentamos un disparador SQL y el valor de las métricas para el mismo:

<pre>CREATE TRIGGER CASO_CUATRO AFTER DELETE ON P FOR EACH ROW WHEN (OLD.CIUDAD='PARIS') BEGIN     DELETE FROM J WHERE J.J# IN         (SELECT J# FROM SPJ WHERE SPJ.S# IN         (SELECT S# FROM S WHERE S.SNOMBRE = 'CORONA')); END;</pre>	<table border="1"> <tbody> <tr> <td>NO= 1</td> </tr> <tr> <td>NS=3</td> </tr> </tbody> </table>	NO= 1	NS=3
NO= 1			
NS=3			

Fig. 2. Ejemplo de disparador y valor de las métricas

## Validación empírica de las métricas

A la hora de trabajar con métricas, hay que ir más allá de la pura definición. Es necesario someter a las métricas a procesos de verificación formal y validación

<sup>1</sup> Se excluye aquí la posibilidad de que la acción sea un programa PL/SQL ya que en ese caso podría directamente aplicarse cualquier métrica tradicional como la complejidad ciclomática propuesta por McCabe (1976).

empírica (ver figura 3). En general este proceso será iterativo y nos permitirá la redefinición de las métricas propuestas en caso de ser necesario hasta poder obtener métricas que nos ayuden a medir lo que realmente se desea, y que además se haya demostrado su validez tanto desde un punto de vista formal como empírico.

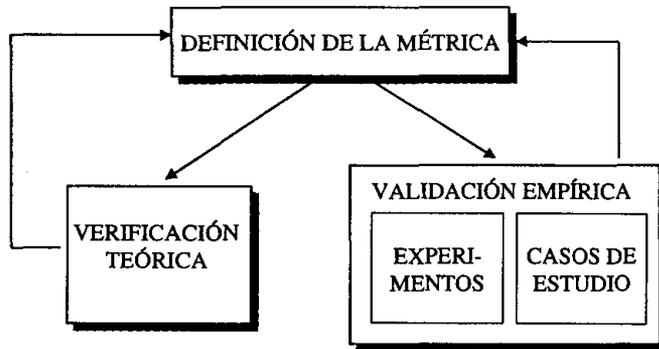


Fig. 3. Método utilizado para la definición de métricas

En esta sección, presentamos los estudios empíricos llevados a cabo para demostrar la utilidad de las métricas propuestas. Se presentan dos estudios cuya finalidad es la misma (trabajan con las mismas hipótesis) pero que difieren entre sí en la parte de la acción de los disparadores. La realización de réplicas resulta imprescindible ya que de esta forma se pueden ratificar los resultados obtenidos (Basili, 1999).

En ambos casos, los estudios han sido realizados siguiendo el método experimental aplicado en la Ingeniería del Software (Bourque and Côte, 1991; Pfleeger, 1995; Dolado y Fernández, 2000) y que consiste en los siguientes pasos: formulación de las hipótesis, descripción de los sujetos, descripción del material utilizado para la realización del experimento, diseño del experimento e interpretación de los resultados. A continuación presentamos toda esta información relativa a los experimentos desarrollados, teniendo en cuenta que siempre que no se diga lo contrario, la información fue idéntica en ambos casos y que el objetivo de ambos experimentos es demostrar el impacto de los factores número de sentencias (NS) y número de operandos (NO) en la comprensibilidad de las reglas.

#### Hipótesis iniciales

- Hipótesis nula: La complejidad de los disparadores de la base de datos activa no está influida por los diferentes valores de las métricas.
- Hipótesis alternativa 1: La complejidad de los disparadores de la base de datos activa depende del valor de la métrica NO

- Hipótesis alternativa 2: La complejidad de los disparadores de la base de datos activa depende del valor de la métrica NS
- Hipótesis alternativa 3: La complejidad de los disparadores de la base de datos activa depende de la combinación de los valores de las métricas NO y NS.

### Sujetos

En ambos experimentos los sujetos que realizaron el experimento fueron alumnos universitarios de Informática. En el primer experimento los alumnos eran de la Universidad de Castilla-La Mancha y pertenecían al quinto curso de carrera. En el momento de la realización del experimento, cursaban una asignatura específica de Bases de Datos Avanzadas en la que habían visto conceptos sobre actividad en bases de datos. En el segundo, los alumnos eran de la Universidad Rey Juan Carlos, y aunque pertenecían al tercer curso de carrera, también estaban cursando una asignatura del mismo tipo, por lo que en ambos casos, el conocimiento de los sujetos en bases de datos activas era equiparable. En el primer caso el experimento fue realizado por 19 alumnos aunque sólo pudo trabajarse con los resultados de 18 de ellos teniendo que rechazar uno por no ser correcto. En el segundo caso, el experimento fue desarrollado por 22 sujetos de los que luego se seleccionaron 14 por idénticos motivos a los del primer caso.

### Materiales del experimento

El material en ambos casos consiste en dos documentos (ver apéndice 1). El primero incluye el esquema de la base de datos, el diagrama entidad/interrelación, y el conjunto de tuplas para cada tabla. El segundo se compone de los cuatro tests que los sujetos debían realizar. El test consiste en averiguar el estado final de la base de datos después de la activación de las reglas.

Existía una pequeña variación en la parte de la acción de los disparadores de los diferentes tests, ya que no se realizó el mismo experimento en ambos casos. La diferencia estribó en el tipo de operaciones que se incluyeron. Mientras en el primer caso estas operaciones podían ser tanto operaciones de borrado simples como operaciones de borrado con subconsultas, en la segunda ocasión las operaciones se limitaron a ser borrados simples de tuplas. La idea de esta variación fue averiguar si el tipo de operación utilizada en la acción (y, por tanto, capturada por la métrica) influía y podía ser necesario hacer una distinción del tipo de operación en la métrica.

Cada sujeto disponía de tiempo limitado para realizar los tests, registrando la hora de comienzo y terminación para cada uno de los test. Sólo fueron contabilizados los tests correctamente contestados.

## Diseño experimental

Teniendo en cuenta las hipótesis, los experimentos debían trabajar con dos factores o variables independientes, el número de operandos (NO) y el número de sentencias (NS), lo cual nos llevó a un modelo factorial y más concretamente a un diseño cruzado en el que cada nivel de cada factor se combina con todos los niveles del otro factor. En nuestro caso, trabajamos con dos niveles para cada factor, concretamente cada factor podía tomar los niveles 1 y 3. Además, y debido a que en cada casilla había el mismo número de observaciones (número de sujetos que respondieron correctamente), nuestro modelo es balanceado. Este diseño se muestra en la figura 4, donde cada celda de la tabla es un par (NS, NO).

		Factor B (NS)	
		LOW	HIGH
Factor A	LOW (NO)	1,1	1,3
	HIGH	3,1	3,3

Fig. 4. Diseño cruzado para los experimentos

## Variables Independientes

Las variables independientes de nuestro experimento eran el número de operandos (NO) y el número de sentencias (NS). Cada una de estas variables independientes podía tener dos niveles, 1 y 3.

## Variables Dependientes

En nuestros experimentos la variable dependiente es el tiempo. Como el objetivo de los experimentos era medir la complejidad, se decidió dar a los sujetos tanto tiempo como fuese necesario para la realización de cada uno de los tests. Así sólo se consideraron como válidos aquellos tests que habían sido correctamente respondidos, trabajando, por lo tanto, con los tiempos que se habían invertido en su realización. Esta información fue proporcionada por los mismos sujetos, que debieron encargarse de apuntar para cada test la hora de comienzo y de finalización.

## Variables Controladas

Se intentó reducir al mínimo la variabilidad entre los participantes haciendo que todos tuvieran los mismos conocimientos previos de bases de datos activas. Además, todos los participantes realizaron las mismas pruebas. Asimismo, a la hora de diseñar

los disparadores se tuvo en cuenta que todos ellos tuvieran una serie de características comunes: granularidad. FOR EACH ROW, tiempo de activación: AFTER, condición: SIEMPRE CIERTA, operador: OR.

### Procedimiento

Los cuatro tests fueron realizados seguidos durante una hora de tiempo. Los sujetos tuvieron la responsabilidad de anotar los tiempos de comienzo y finalización de cada uno de los tests. Además, y aunque todos ellos realizaron los mismos tests, no les fueron entregados a todos en el mismo orden.

Una vez corregidos los tests, sólo se seleccionaron los que estaban correctamente contestados para la obtención de los resultados de los experimentos.

### Resultados del experimento

Tres son los factores a tener en cuenta para el análisis de datos: la naturaleza de los datos, la finalidad del experimento y, el tipo de diseño experimental utilizado (Pfleeger, 1995). Teniendo en cuenta, como ya hemos mencionado, que tenemos un modelo factorial en el que queremos averiguar la interacción entre dos factores, la técnica más conveniente para obtener los resultados es el F-estadístico (Rohatgi, 1976).

El poder de los tests estadísticos radica en tres componentes:  $\alpha$  (margen de error), el tamaño del efecto que se investiga y el número de sujetos. Como el efecto del tamaño y el número de sujetos son constantes, la única opción que nos queda para aumentar el poder del test es incrementar el valor de  $\alpha$ . En este sentido, en lugar de escoger un valor  $\alpha=0,05$  (95% nivel de confianza) que es más usual, escogemos un valor  $\alpha=0,1$  (90% nivel de confianza).

### Experimento 1:

La tabla 2 muestra los resultados obtenidos a partir de los datos recogidos en el primer experimento.

EXPERIMENTO 1		
Factor de variación	Grados de libertad	F-ratio
NO	1	0.08
NS	1	2.08
INTERACCIÓN	1	1.33
ERROR	68	

Table 2. Resultados del primer experimento.

Con los resultados obtenidos y con  $\alpha=0.1$ , buscamos el valor en tablas de forma que el numerador es 1 (grados de libertad para los tres casos, NS, NO e interacción) y el denominador es 68 (valor que corresponde a los grados de libertad del error y que se calcula como número de datos menos los grados de libertad de los tres casos menos uno). Comparando los valores encontrados en la tabla  $F_{1,68} = 2.79$ , podemos afirmar que:

- ▷ Hipótesis 1. Como  $0.068 < 2.79$ , el valor de NO no afecta a los resultados del experimento. Por tanto esta hipótesis es NO VÁLIDA.
- ▷ Hipótesis 2. Como  $2.08 < 2.79$ , el valor de NS no afecta a los resultados del experimento. Por tanto esta hipótesis es NO VÁLIDA.
- ▷ Hipótesis 3. Como  $1.33 < 2.79$ , la interacción de ambas métricas no afecta a los resultados del experimento. Por tanto esta hipótesis es NO VÁLIDA.

Finalmente, en vista de estos resultados, concluimos que la hipótesis que confirma el experimento es la hipótesis nula y, por lo tanto, parece ser que ni el número de operandos de la condición de una regla ni el número de sentencias SQL de la acción de una regla son indicadores de su complejidad.

**Experimento2:**

La tabla 3 muestra los resultados obtenidos a partir de los datos recogidos en el segundo experimento.

EXPERIMENTO 2		
Factor de variación	Grados de libertad	F-ratio
NO	1	1.75
NS	1	2.74
INTERACCIÓN	1	1.22
ERROR	52	

Table 3. Resultados del segundo experimento

Con los resultados obtenidos y con  $\alpha=0.1$ , buscamos el valor en tablas de forma que el numerador es 1 (grados de libertad para los tres casos, NS, NO e interacción) y el denominador es 52 (valor que corresponde a los grados de libertad del error y que se calcula como número de datos menos los grados de libertad de los tres casos menos uno). Comparando los valores encontrados en la tabla  $F_{1,52} = 2.82$ , podemos afirmar que:

- ▷ Hipótesis 1. Como  $1.75 < 2.82$ , el valor de NO no afecta a los resultados del experimento. Por tanto esta hipótesis es NO VÁLIDA.
- ▷ Hipótesis 2. Como  $2.74 < 2.82$ , el valor de NS no afecta a los resultados del experimento. Por tanto esta hipótesis es NO VÁLIDA.

- ⇒ Hipótesis 3. Como  $1.22 < 2.82$ , la interacción de ambas métricas no afecta a los resultados del experimento. Por tanto esta hipótesis es NO VÁLIDA.

Como en el primer experimento, concluimos que la hipótesis que confirma el experimento es la hipótesis nula y, por lo tanto, parece ser que ni el número de operandos de la condición de una regla ni el número de sentencias SQL de la acción de una regla son indicadores de su complejidad

Resulta evidente que esta misma conclusión puede también considerarse como la de ambos experimentos aunque por ser este un primer estudio no pueden darse estos resultados como definitivos aunque sí como orientativos.

Ambos experimentos parecen confirmar el hecho de que, generalmente, es la interacción entre reglas, más que las reglas en sí mismas, la que influye en la sensación de falta de control por parte del usuario.

Sin embargo, y debido al estrecho margen que existe en ambos experimentos entre el valor para la segunda hipótesis respecto al valor en tablas, podría hacerse necesario un nuevo experimento en el que participaran un mayor número de sujetos o en el que la diferencia entre los niveles del factor correspondiente a la métrica NS fuera mayor para poder de esta forma confirmar o no las conclusiones que se derivan de este primer experimento.

## Conclusiones

Las bases de datos juegan un papel cada vez más importante en las sociedades de la información. Esta creciente dependencia de este recurso debe llevar pareja una preocupación por la calidad de los datos. En este trabajo hemos propuesto y validado dos métricas para evaluar la complejidad intra-regla en las bases de datos activas: el número de operandos para la parte condición, y el número de sentencias SQL para la parte acción. Los resultados de una primera validación han arrojado que ninguno de las dos métricas son significativas para evaluar la complejidad de las reglas ECA.

Sin embargo, sería conveniente realizar el experimento con profesionales y con datos reales para poder, de esta forma, contrastar los resultados obtenidos, ya que los experimentos controlados en laboratorio presentan deficiencias por lo que resulta necesario combinar estos experimentos controlados con casos de estudio (Basili, 1999).

Estas métricas para bases de datos activas, están siendo complementadas con otras para bases de datos relacionales (Calero et al., 2000) y objeto-relacionales (Calero et al., 1999).

## Agradecimientos

Esta investigación forma parte del proyecto MANTICA, parcialmente financiado por la CICYT y la Unión Europea (CICYT-1FD97-0168)

Queremos agradecer a la Doctora Esperanza Marcos y a sus alumnos de la Universidad Rey Juan Carlos la ayuda prestada para la realización del segundo experimento.

## Bibliografía

- Basili, V. R. (1999). *Using experiments to build a body of knowledge*. Conference at the Polytechnical University, Madrid. Spain.
- Bourque P. y Côte V. (1991). An experiment in Software Sizing with structured analysis metrics. *J. Systems Software*, 15, 159-172.
- Briand, L.C., Differding, C.M. y Rombach, D. (1996). Practical Guidelines for Measurement-Based Process Improvement. *Software Process-Improvement and Practice 2*, 253-280.
- Calero, C., Piattini, M., Ruiz, F. y Polo, M. (1999) Validation of metrics for Object-Relational Databases, *International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (ECOOP99)*, Lisbon (Portugal), 14-18 June
- Calero, C., Piattini, M., Genero, M., Serrano, M. y Caballero, I. (2000). Metrics for Relational Databases Maintainability, *UKAIS2000*, Cardiff, UK. pp. 109-119
- Díaz, O., Piattini, M. y Calero, C. (2000). Measuring active databases maintainability. Aceptado para su publicación en *Information Systems Journal*
- Dolado, J.J. y Fernández, L. (2000). *Medición para la gestión en la Ingeniería del Software*. Ed. RA-MA. Madrid
- Fenton, N. y Pfleeger, S. L. (1997). *Software Metrics: A Rigorous Approach 2<sup>nd</sup>*. edition. London, Chapman & Hall.
- Henderson-Sellers, B. (1996). *Object-oriented Metrics - Measures of complexity*. Prentice-Hall, Upper Saddle River, New Jersey.
- Kitchenham, B.A. y Linkman, S.J. (1990). Design metrics in practice. *Information and Software Technology*, 32 (4), 304-310.
- Kotz, A. M. (1994). Active database functionality in a real-world banking environment. *Dagstuhl Seminar on Active DBMS*.
- Li, H.F. y Chen, W.K. (1987). An empirical study of software metrics. *IEEE Trans. on Software Engineering*, 13 (6), 679-708.
- McCabe, T.J. (1976). A complexity measure. *IEEE Trans. Software Engineering* 2(5), 308-320.
- Paton, M. y Díaz, O. (1998). Active Databases Systems. *ACM Computer Surveys*, 31(1), 63-103.
- Pfleeger S.L. (1995). Experimental design and analysis in software engineering. *Annals of Software Engineering*, 219-253, JC Baltzer AG, Science Publishers.
- Pfleeger, S. L. (1997). "Assessing Software Measurement". *IEEE Software*. March/April, pp. 25-26.
- Rohatgi, V.K. (1976). *An introduction to probability theory and mathematical statistics*. Wiley Series in Probability and mathematical statistics.
- Widom, J. y Cori, S. (eds.) (1996). *Active Database Systems*. Morgan Kaufmann.

### Apéndice 1.

Material entregado a cada alumno y plantilla de respuestas de uno de los tests del experimento:

**CREACIÓN DE LAS TABLAS**

```

CREATE TABLE S
(S# CHAR(2),
SNOMBRE VARCHAR(15),
SITUACION NUMBER,
CIUDAD VARCHAR(15),
PRIMARY KEY (S#));

CREATE TABLE P
(P# CHAR(2),
PNOMBRE VARCHAR(15),
COLOR VARCHAR(15),
PESO NUMBER,
CIUDAD VARCHAR(15),
PRIMARY KEY (P#));

CREATE TABLE J
(J# CHAR(2),
JNOMBRE VARCHAR(15),
CIUDAD VARCHAR(15),
PRIMARY KEY (J#));

CREATE TABLE SPJ
(S# CHAR(2),
P# CHAR(2),
J# CHAR(2),
CANTIDAD NUMBER,
PRIMARY KEY(S#,P#,J#));
    
```

NOMBRE: \_\_\_\_\_

HORA COMIENZO (hh:mm): \_\_\_\_\_

DISPARADOR:

```

CREATE TRIGGER UNO
AFTER DELETE ON S
FOR EACH ROW
WHEN (OLD.CIUDAD=LONDRES)
BEGIN
DELETE FROM SPJ WHERE SPJ.S# = OLD.S#;
END UNO;

EVENTO
DELETE FROM S WHERE S# = 'S1';

SALIDA:


| TABLA S | TABLA P | TABLA J | TABLA SPJ |
|---------|---------|---------|-----------|
|         |         |         |           |


```

HORA FIN (hh:mm): \_\_\_\_\_

**CONTENIDO DE LAS TABLAS**

**Tabla S**

S#	SNOMBRE	SITUACION	CIUDAD
S1	SALAZAR	20	LONDRES
S2	JAIMES	10	PARIS
S3	BERNAL	30	PARIS
S4	CORONA	20	LONDRES
S5	ALDANA	30	ATENAS

**Tabla P**

P#	PNOMBRE	COLOR	PESO	CIUDAD
P1	TUERCA	ROJO	12	LONDRES
P2	PERNO	VERDE	17	PARIS
P3	BIRLO	AZUL	17	ROMA
P4	BIRLO	ROJO	14	LONDRES
P5	LEVA	AZUL	12	PARIS
P6	ENGRANE	ROJO	19	LONDRES

**Tabla J**

J#	JNOMBRE	CIUDAD
J1	CLASIFICADOR	PARIS
J2	PERFORADORA	ROMA
J3	LECTORA	ATENAS
J4	CONSOLA	LONDRES
J5	COMPAGINADOR	OSLO
J6	TERMINAL	LONDRES
J7	CINTA	LONDRES

**Tabla SPJ**

S#	P#	J#	CANTIDAD
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S3	P5	J1	100
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P2	J5	500
S5	P5	J7	100
S5	P1	J4	100
S5	P4	J4	200
S5	P5	J4	800
S5	P6	J4	400
S5	P6	J4	500