

## Patrocinadores



UNIVERSIDAD DE ZARAGOZA



## Entidades Organizadoras

- Adaspain.
- Asociación de Enseñantes Universitarios de la Informática (AENUU).
- Asociación de Técnicos Informáticos (ATI).
- Asociación Española para la Inteligencia Artificial (AEPPIA).
- Asociación para la Interacción Persona-Ordenador (AIPO).
- Asociación para el Desarrollo de la Informática Educativa (ADIE).
- Ayuntamiento de Zaragoza.
- Capítulo Español de la IEEE Computational Intelligence Society.
- Comité Español de Automática (CEA).
- Conferencia de Decanos y Directores de Informática (CODDI) de las Universidades Españolas.
- Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza.
- European Society for Fuzzy Logic and Technology (EUSFLAT).
- Federación de Asociaciones de Ingenieros en Informática (AI2).
- W3C España (World Wide Web Consortium).
- Programa Nacional de Tecnologías Informáticas - Dirección General de Investigación, Ministerio de Educación y Ciencia.
- Red Española de Metaheurísticas.
- Red Española de Minería de Datos y Aprendizaje.
- Sección Española de la European Association for Computer Graphics (EUROGRAPHICS).
- Sociedad de Arquitectura y Tecnología de Computadores (SARTECO).
- Sociedad de Ingeniería del Software y Tecnologías de Desarrollo del Software (SISTEDES).
- Universidad de Zaragoza.

ISBN: 978-84-9732-595-0

CEDI 2007 XII Jornadas de Ingeniería del Software y Bases de Datos | JISBD'07 |

# CEDI 2007

II CONGRESO ESPAÑOL  
DE INFORMÁTICA  
ZARAGOZA SPAINI

AUDITORIO PALACIO DE CONGRESOS  
11 AL 14 DE SEPTIEMBRE DE 2007

## XII Jornadas de Ingeniería del Software y Bases de Datos

| JISBD'07 |



EDITOR

Xavier Franch

**CEDI 2007**  
II CONGRESO ESPAÑOL  
DE INFORMÁTICA  
Nuevos retos  
científicos y tecnológicos  
en Ingeniería Informática  
**ZARAGOZA SPAIN**  
DEL 11 AL 14 DE SEPTIEMBRE

---



# ACTAS DE LAS XII JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS

**EDITOR**

Xavier Franch

**PATROCINA**

**INTERSYSTEMS**

**COLABORA**

**THOMSON**  
—★—™



**ACTAS DE LAS XII JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS (JISBD'07)**

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier otro medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros medios, sin el permiso previo y por escrito de los titulares del Copyright.

Derechos reservados ©2007 respecto a la primera edición en español, por LOS AUTORES  
Derechos reservados ©2007 International Thomson Editores Spain, S.A.

Magallanes, 25; 28015 Madrid, ESPAÑA  
Teléfono 91 4463350  
Fax: 91 4456218  
clientes@parainfo.es

ISBN: 978-84-9732-595-0  
Depósito legal: M-

Maquetación: Los Editores  
Coordinación del proyecto: @LIBROTEX  
Portada: Estudio Dixi  
Impresión y encuadernación: FER Fotocomposición, S. A.

IMPRESO EN ESPAÑA-PRINTED IN SPAIN

## Comité Ejecutivo

### **Presidente del Comité de Programa**

*Xavier Franch (Universitat Politècnica de Catalunya)*

### **Secretario de la Comisión Permanente**

*Mario Piattini (Universidad de Castilla-La Mancha)*

### **Coordinadora de Tutoriales**

*Ana M. Moreno (Universidad Politécnica de Madrid)*

### **Coordinador de Talleres**

*Vicente Pelechano (Universidad Politécnica de Valencia)*

### **Coordinador de Demostraciones**

*Antonio Vallecillo (Universidad de Málaga)*

### **Coordinador de la Sesión de Divulgación de Trabajos Relevantes ya Publicados**

*Oscar Díaz (Universidad del País Vasco)*

### **Composición y Maquetación de Actas**

*Jordi Marco (Universitat Politècnica de Catalunya)*

### **Organización y Relaciones con CEDI 2007**

*Fran J. Ruiz (Universidad de Zaragoza)*

*M. Elena Gómez (Universidad de Zaragoza)*

*Javier Tuya (Universidad de Oviedo)*

## Comité Organizador

### **Presidente del CEDI**

*Alberto Prieto (Universidad de Granada)*

### **Presidente del Comité Científico**

*Juan J. Moreno (Universidad Politécnica de Madrid)*

### **Presidente del Comité Organizador CEDI 2007**

*Victor Viñals (Universidad de Zaragoza)*

### **Coordinador de Actividades Plenarias CEDI 2007**

*José Duato (Universidad Politécnica de Valencia)*

### **Secretario del CEDI 2007**

*José A. Castellanos (Universidad de Zaragoza)*

*José A. Bañares (Universidad de Zaragoza)*

## Comité de Programa

Alberto Abelló, Univ. Polit. Catalunya	Jon Iturrioz, Univ. País Vasco
Silvia Abrahão, Univ. Polit. Valencia	Natalia Juristo, Univ. Polit. Madrid
Jesus Aguilar, Univ. Sevilla	Patricio Letelier, Univ. Polit. Valencia
José Aldana, Univ. Málaga	Antonia Lopes, Univ. Lisboa
Bárbara Álvarez, Univ. Polit. Cartagena	Adolfo Lozano, Univ. Extremadura
María J. Aramburu, Univ. Jaume I	Esperanza Marcos, Univ. Rey Juan Carlos
João Araújo, Univ. Nova de Lisboa	Eduardo Mena, Univ. Zaragoza
Orlando Belo, Univ. do Minho	Ana Moreira, Univ. Nova de Lisboa
Rafael Berlanga, Univ. Jaume I	Juan J. Moreno, Univ. Polit. Madrid
Pere Botella, Univ. Polit. Catalunya	Juan M. Murillo, Univ. Extremadura
Nieves Brisaboa, Univ. Coruña	Oscar Pastor, Univ. Polit. Valencia
Isabel S. Brito, Inst. Polit. Beja	Antonio Polo, Univ. Extremadura
Coral Calero, Univ. Castilla-La Mancha	Carme Quer, Univ. Polit. Catalunya
Carlos Canal, Univ. Málaga	Celia Ramos, Univ. Algarve
José M. Cavero, Univ. Rey Juan Carlos	Isidro Ramos, Univ. Polit. Valencia
Matilde Celma, Univ. Polit. Valencia	José Riquelme, Univ. Sevilla
Rafael Corchuelo, Univ. Sevilla	Antonio Rito, Univ. Técnica de Lisboa
Dolors Costal, Univ. Polit. Catalunya	Antonio Ruíz, Univ. Sevilla
Yania Crespo, Univ. Valladolid	Francisco Ruíz, Univ. Castilla-La Mancha
Oscar Dieste, Univ. Polit. Madrid	José Samos, Univ. Granada
Javier Dolado, Univ. País Vasco	Fernando Sánchez, Univ. Extremadura
João Falcão e Cunha, Univ. Porto	Juan Sánchez, Univ. Polit. Valencia
Pablo de la Fuente, Univ. Valladolid	Ernest Teniente, Univ. Polit. Catalunya
Lidia Fuentes, Univ. Málaga	Miguel Toro, Univ. Sevilla
Mario Gaspar da Silva, Univ. Lisboa	Ambrosio Toval, Univ. Murcia
Marcela Genero, Univ. Castilla-La Mancha	Juan C. Trujillo, Univ. Alicante
Cristina Gómez, Univ. Polit. Catalunya	Javier Tuya, Univ. Oviedo
Jaime Gómez, Univ. Alicante	Belén Vela, Univ. Rey Juan Carlos
Alfredo Goñi, Univ. País Vasco	Cristina Vicente, Univ. Polit. Cartagena
Juan Hernández, Univ. Extremadura	

## Comité Asesor para la Selección de Trabajos de Prestigio

Oscar Díaz (Presidente), Univ. País Vasco	Neil A.M. Maiden, City Univ. London
Alan Davis, Univ. of Colorado	Timos Sellis, Nat. Technical Univ. Athens

## Revisores Adicionales

César J. Acuña  
Amaia Aguirregoitia  
Diego Alonso  
David Benavides  
Jordi Cabot  
Paloma Cáceres  
Javier Cámara  
Dante Carrizo  
Pedro J. Clemente  
Jose M. Conejero  
Javier Cubo  
Norberto Díaz  
Amador Durán  
Sergio España  
Mauricio Espinoza  
Ismael Etxeberria  
Antonio Fariña  
Raul Fernandez  
L. Fredlund  
Antonielly Garcia  
Antonio Cesar Gómez  
Ángel Herranz  
Sergio Ilarri  
Miguel Ángel Laguna  
Maria Lencastre  
Marta López  
Francisco Javier Lucas  
María Esperanza Manso  
Julio Mariño  
José Manuel Marqués  
Francisco Martínez  
Jorge Martínez

Miguel Ángel Martínez  
Fernando Molina  
Ana M. Moreno  
Elena Navarro  
Ismael Navas  
Isabel Nepomuceno  
Juan A. Nepomuceno  
Joaquín Nicolás  
Guadalupe Ortiz  
Juan Angel Pastor  
Joaquin Peña  
Jenifer Pérez  
Juan Manuel Pérez  
Beatriz Pontes  
Álvaro Prieto  
Antonia M. Reina  
Domingo Savio Rodríguez  
Roberto Rodríguez  
Oscar Romero  
Fran J. Ruiz  
Angeles Saavedra  
Gwen Salaün  
Pedro Sánchez  
André L. Santos  
Diego Seco  
Jesús Serrano  
Encarna Sosa  
Toufik Taibi  
Raquel Trillo  
José Antonio Troyano  
Juan Manuel Vara

## **Sistema Automático de Revisión**

*Quercus Software Engineering Group*

Jose Javier Berrocal Universidad de Extremadura

Conferencia auspiciada por



## Prólogo

Respondiendo a su cita anual, las XII Jornadas de Ingeniería del Software y Bases de Datos (JISBD) se han celebrado en Zaragoza, entre el 11 y el 14 de septiembre de 2007. Las Jornadas representan un punto de encuentro de la comunidad investigadora en ingeniería del software y en bases de datos. En sus inicios se celebraron dos eventos diferenciados, las Jornadas de Ingeniería del Software y las Jornadas sobre Investigación y Docencia en Bases de Datos. Posteriormente, en 1999, ambos eventos se unificaron en uno solo, reflejando la interrelación existente entre estas disciplinas. En esta duodécima edición, las Jornadas han constituido, una vez más, un punto de encuentro en el que profesionales y académicos de España, Portugal y Latinoamérica, de ambos campos, han podido compartir experiencias y resultados entre distintos grupos de investigación, desarrollo e innovación tecnológica.

Actualmente, JISBD es un evento auspiciado por Sociedad de Ingeniería del Software y Tecnologías de Desarrollo de Software (SISTEDES, <http://www.sistedes.org>). Entre los fines de dicha organización destacan el de promover la investigación, la innovación y la transferencia de tecnología entre los distintos agentes involucrados en el avance las tecnologías del Software y el de fomentar actividades con otras asociaciones nacionales e internacionales con fines similares, consiguiendo así proporcionar una mayor visibilidad a la investigación de sus asociados.

Al igual que en 2005, las XII Jornadas de Ingeniería del Software y Bases de Datos se han realizado en el marco del II Congreso Español de Informática (CEDI 2007). Esto ha permitido a los participantes de las Jornadas participar en las diversas actividades de CEDI de interés para toda la comunidad de investigación en Informática, tales como conferencias invitadas y mesas redondas. La celebración cada dos años de JISBD en el marco de CEDI encaja con los objetivos citados de dicha organización.

Este volumen recoge los trabajos seleccionados por el Comité de Programa de JISBD'07. Se recibieron un total de 87 contribuciones de 9 países: España, Portugal, Argentina, Brasil, Chile, Colombia, Cuba, México y Venezuela. Cada contribución fue revisada por tres miembros del Comité de Programa. Posteriormente, se abrió una fase de discusión en la que se debatieron en mayor profundidad algunos trabajos y eventualmente se pidieron revisiones adicionales para ellos; asimismo, algunos trabajos se aceptaron condicionalmente, pendientes de verificar que la versión definitiva trataba adecuadamente los comentarios de los revisores; gracias al esfuerzo de los autores, todos estos trabajos fueron finalmente aceptados. Como resultado de todo el proceso, se configuró un programa compuesto por 30 artículos. Adicionalmente, se seleccionaron 5 trabajos más para su presentación como artículos cortos. Además, en esta edición de JISBD se recogió la posibilidad de presentar trabajos ya publicados en foros de prestigio reconocido. Se seleccionaron 4 artículos de esta modalidad. Finalmente, destacamos la celebración de una sesión para la presentación de herramientas, cuya convocatoria tuvo una acogida excelente por parte de la comunidad de JISBD, de manera que en dicha sesión se programaron un total de 19 demostraciones de herramientas.

El día previo a la conferencia, se organizaron un total de 7 talleres y un tutorial. Estos eventos están ganando importancia a cada nueva edición de JISBD y en el caso de los talleres, están creando sus propias comunidades con intereses más específicos. Algunos talleres ya están plenamente consolidados y llegan a acumular hasta un total de 8 ediciones. Cabe destacar que a partir de este año, las actas de los talleres se recogen en una publicación única en formato electrónico, con el soporte de SISTEDES, para potenciar la difusión de los trabajos presentados.

En referencia al programa, mencionar la participación de dos conferenciantes invitados de reconocido prestigio, siguiendo la pauta de ediciones anteriores. La primera conferencia impartida por Stephen Mellor, miembro del Object Management Group, y con un largo historial en la formulación de métodos para el análisis orientado a objetos. La segunda conferencia a cargo del profesor

John Mylopoulos, que posee igualmente una dilatada experiencia en diversos ámbitos de la ingeniería del software. La presencia de estos dos investigadores representó un elemento importante en el programa de las Jornadas.

Quisiera destacar un hecho que no por obvio, deja de ser merecedor de mención. La celebración de un evento de las características de JISBD, con una participación cada vez más numerosa y consolidada, y con unas exigencias de calidad que se van incrementando en cada edición, no podría realizarse sin la dedicación totalmente desinteresada de un gran número de personas. Desde el punto de vista científico, el trabajo en equipo desarrollado por los miembros del Comité Ejecutivo, en cuyo seno se han debatido los temas más candentes en la configuración de la oferta científica del congreso; y por supuesto la ardua y puntual labor de revisión efectuada por los miembros del Comité de Programa y los revisores adicionales. Desde el punto de vista organizativo, destacar la gran dedicación de los miembros del Comité Ejecutivo responsables de las tareas de enlace con CEDI, y la labor del Grupo Quercus de Ingeniería del Software de la Universidad de Extremadura, quienes han estado a cargo de todo el sistema de recepción y revisión de artículos. También deseo agradecer el soporte recibido por las entidades patrocinadoras y colaboradoras, y en especial la labor de respaldo de SISTEDES, tanto por lo que se refiere a apoyo logístico como a tareas de difusión, como ya se ha comentado. Y por último, especialmente, a los autores de los trabajos enviados a JISBD'07, en definitiva son ellos los que hacen posible la celebración del evento.

Finalmente, desear que el volumen que ahora tienes en tus manos, y que refleja el estado del arte en la investigación en Ingeniería del Software y Bases de Datos en la comunidad de habla hispana y portuguesa, sea de utilidad para tu trabajo.

Zaragoza, Septiembre 2007  
Xavier Franch (editor)

<b>Índice</b>	<b>9</b>
---------------	----------

## Índice

### CONFERENCIAS INVITADAS

<b>Creativity, Automation and Technology</b>	
<i>Stephen J Mellor</i> . . . . .	15
<b>Goal-Oriented Requirements Engineering</b>	
<i>John Mylopoulos</i> . . . . .	17

### TUTORIAL

<b>Tutorial: Herramientas Eclipse para Desarrollo de Software Dirigido por Modelos</b>	
<i>Cristina Vicente-Chicote y Diego Alonso</i> . . . . .	21

### TRABAJOS RELEVANTES YA PUBLICADOS

<b>Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses</b>	
<i>Eduardo Fernández-Medina, Juan Trujillo, Rodolfo Villarroel y Mario Piattini</i> . . . . .	25
<b>A UML profile for multidimensional modeling in data warehouses</b>	
<i>Sergio Luján-Mora, Juan Trujillo e Il-Yeol Song</i> . . . . .	26
<b>Location-Dependent Queries in Mobile Contexts: Distributed Processing Using Mobile Agents</b>	
<i>Sergio Ilarri, Eduardo Mena y Arantza Illarramendi</i> . . . . .	27
<b>Integrating techniques and tools for testing automation</b>	
<i>Macario Polo, Sergio Tendero y Mario Piattini</i> . . . . .	28

### DESARROLLO DE SOFTWARE DIRIGIDO POR MODELOS

<b>Utilidad de las transformaciones modelo-modelo en la generación automática de código</b>	
<i>Javier Luis Cánovas Izquierdo, Óscar Sánchez Ramón, Jesús Sánchez Cuadrado y Jesús García Molina</i> . . . . .	31
<b>Building Ubiquitous Business Process following an MDD approach</b>	
<i>Pau Giner, Victoria Torres y Vicente Pelechano</i> . . . . .	41
<b>A case study on modeling persistence with MDA tools</b>	
<i>Giuliano Luz Pigatti Caliarì y Paulo Sérgio Muniz Silva</i> . . . . .	51

### ALMACENES Y MINERÍA DE DATOS

<b>Ingeniería inversa dirigida por modelos para el diseño de almacenes de datos</b>	
<i>Jose-Norberto Mazón, Enrique Ortega y Juan Trujillo</i>	63
<b>Minería de datos con clustering en espacios multidimensionales mediante modelos conceptuales extendiendo UML</b>	
<i>Jose Zubcoff, Jesús Pardillo y Juan Trujillo</i>	73
<b>Una extensión del metamodelo relacional de CWM para representar Almacenes de Datos Seguros a nivel lógico</b>	
<i>Emilio Soler, Juan Trujillo, Eduardo Fernández-Medina y Mario Piattini</i>	83

### PRUEBAS DEL SOFTWARE

<b>Generación sistemática de pruebas para composiciones de servicios utilizando criterios de suficiencia basados en transiciones</b>	
<i>José García-Fanjul, Javier Tuya y Claudio de la Riva</i>	95
<b>Generación automática de objetivos de prueba a partir de casos de uso mediante partición de categorías y variables operacionales</b>	
<i>Javier J. Gutiérrez, María J. Escalona, Manuel Mejías, Jesús Torres y Arturo Torres-Zenteno</i>	105
<b>370.000 bugs del proyecto Debian pueden ser analizados usando btsextract</b>	
<i>Miguel Pérez Francisco y Pablo Boronat Pérez</i>	115

### TECNOLOGÍAS DE BASES DE DATOS

<b>Búsqueda de vecinos en espacios multidimensionales agujereados</b>	
<i>Manuel Barrena, Carlos Pachón y Elena Jurado</i>	125
<b>Indexación dinámica para la recuperación de información basada en búsqueda por similitud</b>	
<i>Nieves R. Brisaboa, Antonio Fariña, Oscar Pedreira y Nora Reyes</i>	134
<b>WCSA: Un autoíndice orientado a palabras para textos en lenguaje natural</b>	
<i>Eduardo Rodríguez, Antonio Fariña, Ángeles S. Places, José R. Paramá y Oscar Pedreira</i>	144

### LÍNEAS DE PRODUCTO. ORIENTACIÓN A ASPECTOS

<b>Variabilidad, Trazabilidad y Líneas de Productos: una Propuesta basada en UML y Clases Parciales</b>	
<i>Miguel A. Laguna y Bruno González-Baixauli</i>	157
<b>Verificación de Modelos Arquitectónicos Orientados a Aspectos</b>	
<i>Jennifer Pérez, Cristóbal Costa, Jose Ángel Carsí e Isidro Ramos</i>	167
<b>Gestión Integral de Requisitos de Seguridad en Líneas de Producto Software</b>	
<i>Daniel Mellado, Eduardo Fernández-Medina y Mario Piattini</i>	177

## REQUISITOS. METAMODELADO EN MEDICIÓN

<b>Una metodología para elicitación de requisitos en proyectos GSD</b> <i>Gabriela N. Aranda, Aurora Vizcaíno, Alejandra Cechich, Mario Piattini y Juan Pablo Soto</i>	191
<b>Una Aproximación de Metamodelado para la Evaluación de Calidad en Procesos de Desarrollo Web</b> <i>Cristina Cachero, Emilio Insfran, Silvia Abrahão y Geert Poels</i>	201
<b>Marco de Trabajo basado en MDA para la Medición Genérica del Software</b> <i>Beatriz Mora, Félix García, Francisco Ruiz, Mario Piattini, Artur Boronat, Abel Gómez, José Á. Carsí e Isidro Ramos</i>	211

## MODELIZACIÓN CONCEPTUAL DE DATOS

<b>Definición, importancia y especificación en UML de las restricciones de integridad constante y permanente</b> <i>Raquel Pau y Antoni Olivé</i>	223
<b>Modelado de Aplicaciones Web Reactivas al Usuario</b> <i>Irene Garrigós y Jaime Gómez</i>	232
<b>Towards Integration of Access Control in the Hypermedia Development Process</b> <i>Daniel Sanz, Paloma Díaz e Ignacio Aedo</i>	242

## ARQUITECTURAS SOFTWARE

<b>Diseño de Sistemas Groupware sobre una Arquitectura centrada en Servicios Cooperativos: Ágora</b> <i>Miguel A. Martínez-Prieto, Pablo de la Fuente y Carlos E. Cuesta</i>	255
<b>Una Propuesta de Libro Electrónico basada en Composición de Responsabilidades sobre la Estructura Lógica</b> <i>Miguel A. Martínez-Prieto, Pablo de la Fuente, Jesús Vegas y Joaquín Adiego</i>	265
<b>Recuperación y procesado de datos biológicos mediante Ingeniería Dirigida por Modelos</b> <i>Abel Gómez, Artur Boronat, Claudia Täubner, Jose Á. Carsí, Isidro Ramos y Silke Eckstein</i>	275

## MODELOS DE CALIDAD

<b>Evaluando la Calidad de los Datos en Portales Web</b> <i>Angélica Caro, Coral Calero y Mario Piattini</i>	287
<b>Una propuesta de un modelo conceptual de calidad de almacenes de datos</b> <i>Manuel Serrano, Rafael Romero, Jose-Norberto Mazón, Juan Trujillo y Mario Piattini</i>	297
<b>Evaluación de los niveles de calidad en las transformaciones de modelos basado en el estudio de factores de éxito</b> <i>Alejandro Gómez, Gustavo Muñoz y Juan Carlos Granja</i>	307

## PROCESOS

<b>Técnica de Mejora del Mantenimiento Software Basada en Valor</b> <i>Daniel Cabrero, Javier Garzás y Mario Piattini</i> . . . . .	317
<b>Modelo para la Implementación de Mejora de Procesos en Pequeñas Organizaciones Software</b> <i>Francisco J. Pino, Juan C. Vidal, Félix Garcia y Mario Piattini</i> . . . . .	326
<b>Especificación de Procesos de Negocio Seguros a través de una extensión de UML 2.0</b> <i>Alfonso Rodríguez, Eduardo Fernández-Medina, Mario Piattini y Juan Trujillo</i> . . . . .	336

## ARTÍCULOS CORTOS

<b>Eficacia del método ELVIRA - Relato de un experimento</b> <i>Montse Ereño y Rebeca Cortazar</i> . . . . .	349
<b>Tracking the Evolution of Feature Oriented Product Lines</b> <i>Salvador Trujillo, Gentzane Aldekoa y Goiuri Sagardui</i> . . . . .	355
<b>Transformaciones QVT para la obtención de Clases de Análisis a partir de un Modelo de Proceso de Negocio Seguro</b> <i>Alfonso Rodríguez, Ignacio García, Eduardo Fernández-Medina y Mario Piattini</i> . . . . .	361
<b>Definición de un Proceso para la Construcción de Refactorizaciones</b> <i>Raúl Marticorena, Carlos López y Yania Crespo</i> . . . . .	367
<b>Combinando Modelos de Procesos y Activos Reutilizables en una Transición poco Invasiva hacia las Líneas de Producto de Software</b> <i>Orlando Avila-García, Antonio Estévez García, E. Victor Sánchez Rebull y José Luis Roda García</i> . . . . .	373

## DEMOSTRACIONES

<b>Generation of Business Process based Web Applications</b> <i>Pau Giner, Victoria Torres y Vicente Pelechano</i> . . . . .	381
<b>PervGT: Herramienta CASE para la Generación Automática de Sistemas Pervasivos</b> <i>Estefanía Serral, Carlos Cetina, Javier Muñoz y Vicente Pelechano</i> . . . . .	383
<b>UMLtoCSP: Una herramienta para la verificación de modelos UML/OCL mediante Constraint Programming</b> <i>Jordi Cabot, Robert Clarisó, Patricia de la Fuente Y Daniel Riera</i> . . . . .	385
<b>MDBE: Una Herramienta Automática para el Modelado Multidimensional</b> <i>Oscar Romero y Alberto Abelló</i> . . . . .	387
<b>MOMENT CASE: Un prototipo de herramienta CASE</b> <i>Abel Gómez, Artur Boronat, Jose Á. Carsí e Isidro Ramos</i> . . . . .	389
<b>Comprobación eficiente de restricciones de integridad en OCL</b> <i>Jordi Cabot y Ernest Teniente</i> . . . . .	391
<b>The MOVA Tool: A Rewriting-Based UML Modeling, Measuring, and Validation Tool</b> <i>Manuel Clavel, Marina Egea y Viviane Torres da Silva</i> . . . . .	393

<b>Demostración de la herramienta AGE (Agile Generative Environment)</b>	
<i>Jesús Sánchez Cuadrado y Jesús García Molina . . . . .</i>	395
<b>ModelSET: Soporte a Edición y Transformaciones de Modelos</b>	
<i>Antonio Estévez García, E. Victor Sánchez Rebull, Francisco Vargas Ruiz, Orlando Avila-García, Adolfo Sánchez-Barbudo Herrera y José Luis Roda García . . . . .</i>	397
<b>PRISMA CASE</b>	
<i>Jennifer Pérez, Cristóbal Costa, Jose A. Carsí e Isidro Ramos . . . . .</i>	399
<b>StateML: modelado gráfico de máquinas de estados y generación de código siguiendo un enfoque MDE</b>	
<i>Cristina Vicente-Chicote, Diego Alonso y Bárbara Álvarez . . . . .</i>	401
<b>V<sup>3</sup> Studio: Un entorno gráfico para el diseño de sistemas basados en componentes siguiendo un enfoque dirigido por modelos</b>	
<i>Cristina Vicente-Chicote, Diego Alonso y Olivier Barais . . . . .</i>	403
<b>REMM-Studio: Un entorno integrado para dar soporte a un enfoque de Ingeniería de Requisitos Dirigido por Modelos</b>	
<i>Cristina Vicente-Chicote, Begoña Moros y Ambrosio Toval . . . . .</i>	405
<b>MORPHEUS: support from AO-Requirements to AO-Software Architecture</b>	
<i>Elena Navarro, Patricio Letelier e Isidro Ramos . . . . .</i>	407
<b>Maudeling: Herramienta de gestión de modelos usando Maude</b>	
<i>José E. Rivera, Francisco Durán, Antonio Vallecillo y J. Raúl Romero . . . . .</i>	409
<b>WebTE: Generación de aplicaciones Web dirigida por modelos</b>	
<i>Santiago Meliá , Jaime Gómez y Jose Luis Serrano . . . . .</i>	411
<b>CE4WEB: Una Herramienta CASE Colaborativa para el Modelado de Aplicaciones con UML</b>	
<i>Víctor M.R. Penichet, María D. Lozano, J.A. Gallud y R. Tesoriero . . . . .</i>	413
<b>MaCMAS CASE Tool Demonstration: MDD-based refinement of Collaboration-Based UML Models</b>	
<i>Joaquín Peña y Antonio Ruiz-Cortés . . . . .</i>	415
<b>FAMA:hacia el análisis automático de modelos de características</b>	
<i>Pablo Trinidad, David Benavides, Sergio Segura y Antonio Ruiz Cortés . . . . .</i>	417

# Una propuesta de un modelo conceptual de calidad de almacenes de datos

Manuel Serrano<sup>1</sup>, Rafael Romero<sup>2</sup>, Jose-Norberto Mazón<sup>2</sup>, Juan Trujillo<sup>2</sup>,  
Mario Piattini<sup>1</sup>

<sup>1</sup> Grupo Alarcos, Universidad Castilla-La Mancha.

{Manuel.Serrano, Mario.Piattini}@uclm.es

<sup>2</sup> Dpto. Lenguajes y Sistemas Informáticos

Universidad de Alicante {romero, jnmazon, jrtrujillo}@dlsi.ua.es

## Resumen

Los almacenes de datos se consideran una parte fundamental de cualquier sistema de apoyo a la decisión, ya que proporcionan los recursos de información adecuados para facilitar la toma de decisiones de una forma integrada. Por lo tanto, la calidad de un almacén de datos es una cuestión clave en cualquier empresa, habiéndose remarcado en varios trabajos la importancia de medir la calidad en el dominio de un almacén de datos. Sin embargo, según nuestra experiencia, las aproximaciones actuales que tratan aspectos de calidad de los almacenes de datos sólo consideran métricas de calidad aisladas, sin formar parte de un modelo de calidad amplio que permita a los diseñadores evaluar la calidad de un almacén de datos de una forma objetiva y sistemática. Con el fin de superar esta limitación, en este artículo, presentamos un modelo de calidad para el dominio de los almacenes de datos. Este modelo permite a los diseñadores evaluar el significado de la calidad en el dominio de los almacenes de datos, y por tanto medir la calidad del modelo conceptual de un almacén de datos de una forma sistemática y objetiva. Hemos utilizado una metodología bien conocida para construir este modelo de calidad, y hemos relacionado los elementos de este modelo de calidad con nuestras métricas previamente definidas y validadas para los modelos conceptuales de almacenes de datos.

## 1. Introducción

Hoy en día, los almacenes de datos han llegado a ser uno de los componentes más importantes de los sistemas de apoyo a la decisión, ya que permiten una administración de datos de manera eficiente y efectiva para proporcionar recursos de información que faciliten la identificación de problemas y oportunidades, la toma de decisiones críticas y la formulación, implementación y evaluación de estrategias. Por lo tanto, la tecnología de almacenes de datos ha llegado a ser una solución adecuada, proporcionando tales recursos en un entorno

donde el aumento de la competencia, los clientes informados y sofisticados, las impredecibles fluctuaciones del mercado y los entornos cambiantes están presionando en exceso a las organizaciones empresariales [1].

Como resultado de su importancia, las compañías han aumentado su inversión en el desarrollo de los almacenes de datos. De hecho, la inversión en esta clase de sistemas es de varios millones de euros al año y aumenta un 20% cada año [2]. Sin embargo, a pesar del potencial de los almacenes de datos y la enorme cantidad de dinero invertida, el éxito no está necesariamente garantizado. Como han afirmado varios autores, un proyecto de almacén de datos es un riesgo real [3] y más del 60% de los almacenes de datos no satisfacen las expectativas del usuario [4]. Por lo tanto, los diseñadores tienen que tratar con las cuestiones de calidad para evitar estos obstáculos y garantizar el éxito de los proyectos de almacenes de datos [5].

Tratar con las cuestiones de calidad no es una tarea fácil, ya que la calidad es un aspecto subjetivo y abstracto para la que no hay una definición universal: generalmente se dice que hay una definición de calidad para cada persona. De esta forma, es muy complejo medir o evaluar la calidad de un producto software de forma objetiva. Se han propuesto varias guías para tratar la complejidad de la calidad en los almacenes de datos pero la mayoría de las veces estas guías no son suficientes, ya que aunque pueden ayudar a los diseñadores en su trabajo, éstos aún deben tomar bastantes decisiones subjetivas, lo que puede conducir a productos "no muy buenos".

Para superar esta subjetividad inherente de la calidad en los proyectos de almacenes de datos, en este trabajo introducimos un modelo de calidad que ayuda a los diseñadores a evaluar la calidad

de los almacenes de datos de una forma objetiva, garantizando de esta forma el éxito en el diseño de un buen almacén de datos. En este artículo, nos centramos en la calidad de los modelos conceptuales de los almacenes de datos y presentamos los primeros pasos hacia la construcción de un modelo de calidad para almacenes de datos. La razón de que este trabajo se centre en la calidad del modelo conceptual resulta obvia: un *buen* diseño puede (o puede que no) llevar a un *buen* sistema, pero un *mal* diseño seguro que lleva a un *mal* producto de baja calidad. Por lo tanto, es importante centrarse en la calidad de los productos desde los primeros pasos de su desarrollo, por ejemplo el modelo conceptual.

El resto de este artículo está estructurado de la siguiente manera: en la siguiente sección se muestra brevemente el estado de la investigación concerniente a los modelos y calidad de los almacenes de datos. En la sección 3, se presenta un método para desarrollar modelos de calidad basados en ISO 9126 [6]. En la sección 4, el método se aplica a la construcción de un modelo de calidad de almacenes de datos. En la última sección se presentan las conclusiones y los trabajos futuros que surgen a partir de este trabajo.

**2. Trabajo relacionado**

Desde que surgiera la definición de almacén de datos, la calidad ha sido una cuestión importante, tanto desde la perspectiva de los datos como del esquema. Este interés en la calidad viene motivado por la importancia de la calidad de la información en los sistemas de apoyo a la decisión.

Se han usado varias técnicas en un intento de mejorar la calidad de los almacenes de datos, tales como modelos de mejora de procesos software. Entre estas técnicas podemos encontrar el modelo de madurez (GMIN) [7] o SPICE [8], que se basan en la creencia de que "*buenos procesos llevan a un buen software*", pero desafortunadamente los buenos procesos no garantizan buen software, sino que solo aumentan la probabilidad de que se pueda obtener un buen software.

En esta búsqueda de la calidad, se han aplicado varios estándares al desarrollo de los almacenes de datos (tales como ISO 9126 y IEEE 1061 [9]), generalmente complementados por

técnicas GQM [10, 11]) para intentar medir la calidad de los almacenes de datos. Sin embargo, estos estándares son demasiado generales y complejos, por lo que no son suficientemente concretos para ser útiles en la evaluación de la calidad de los sistemas.

Con respecto a las medidas de calidad podemos encontrar varias propuestas para medir algunas dimensiones de calidad de los almacenes de datos tales como las métricas propuestas en [12] y [5]. Estas propuestas son buenas aproximaciones a las medidas de almacenes de datos pero no son completas, ya que no son parte de un modelo de calidad que permita a los diseñadores su uso de una forma sistemática y objetiva.

Finalmente, podemos encontrar la propuesta de un modelo de calidad de un almacén de datos, llamado DWQ [13] que intenta asegurar la calidad de los datos *almacenados* para mejorar el uso del almacén de datos; este proyecto tenía algunas dimensiones de calidad, centrándose en la calidad de los datos. No obstante, pensamos que la calidad de los almacenes de datos no debería solo evaluarse en términos de calidad de datos y que la calidad del esquema es tan importante como la calidad de los datos. Por tanto, en este artículo, proponemos un modelo de calidad para modelos conceptuales de almacenes de datos, basado en el estándar ISO 9126.

**3. Método de desarrollo del modelo de calidad**

Desarrollar un modelo de calidad no es una tarea fácil: se deben considerar varios aspectos y el proceso debe llevarse a cabo, por completo, de una forma metodológica y objetiva. Para ello, Franch y Carvallo [14] propusieron un método para construir modelos de calidad basados en el estándar ISO 9126-1 [6]. A continuación se describe brevemente este método.

La metodología propuesta consta de seis pasos y también considera una actividad preliminar (paso 0). Una vista gráfica de la metodología se muestra en la Figura 1. Aunque los pasos parecen ser secuenciales, éstos pueden entrelazarse y repetirse si fuera necesario.

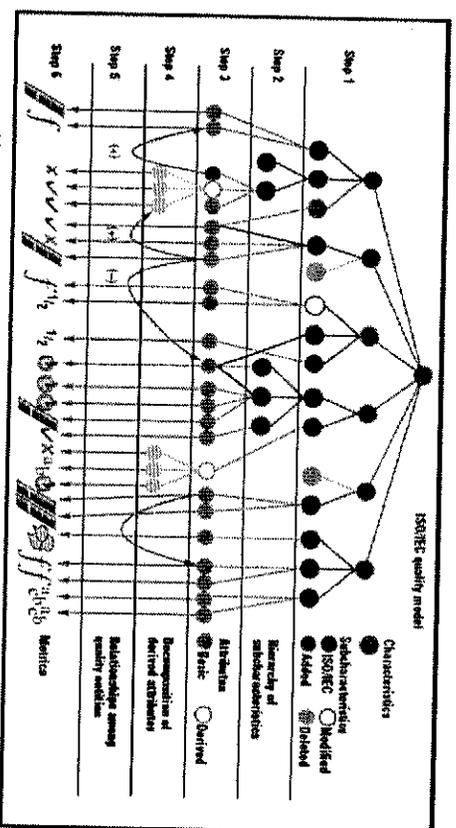


Figura 1. Metodología de seis pasos de Franch and Carvallo [14]

**Paso 0. Definición del dominio**

Antes de la aplicación de la metodología, es necesario examinar y describir cuidadosamente el dominio de interés, con la ayuda de expertos. Con el fin de describir el dominio se recomienda el uso del modelado conceptual para poder realizar un seguimiento de los conceptos relevantes.

**Paso 1. Determinación de las subcaracterísticas de calidad**

La descomposición de características en subcaracterísticas que aparece en el estándar ISO es bastante razonable y debería usarse al menos que durante el análisis del dominio aparecieran buenas razones para no hacerlo. En este caso, se añaden nuevas subcaracterísticas, algunas se pueden refinar y es incluso lícito borrar algunas de las subcaracterísticas que no se aplican al dominio.

**Paso 2. Definición de una jerarquía de subcaracterísticas**

Normalmente, se necesita una nueva descomposición de las subcaracterísticas con respecto a determinados factores. De esta forma, obtenemos una jerarquía de subcaracterísticas.

**Paso 3. Descomposición de las subcaracterísticas en atributos**

Las subcaracterísticas de calidad proporcionan una vista abstracta bastante

comprensible del modelo de calidad. No obstante, necesitamos descomponer estos conceptos abstractos en otros más concretos – los atributos de calidad. Un atributo se corresponde con una determinada característica de los paquetes del dominio. Los atributos deben definirse de forma precisa para clarificar los conceptos de calidad fundamentales que representan y se deben asociar con las características apropiadas.

Como el estándar menciona, no es posible desde un punto de vista práctico, medir todas las subcaracterísticas en la totalidad de un producto software, pero podemos crear una lista de aquellas que son más relevantes.

**Paso 4. Descomposición de los atributos derivados en atributos básicos**

Algunos de los atributos obtenidos en el paso 3 se pueden medir directamente, pero otros pueden ser muy abstractos, requiriendo una nueva descomposición. Por ello, los atributos se clasifican en 'derivados' y 'básicos'. Los atributos derivados se deben descomponer hasta que estén expresados completamente en términos de atributos básicos.

**Paso 5. Establecer relaciones entre las entidades de calidad**

Si deseamos obtener un modelo de calidad completo, debemos establecer las relaciones entre las entidades de calidad. Así, el modelo será más

exhaustivo y las implicaciones de los requerimientos de calidad más claros.

Podemos clasificar las relaciones entre entidades en tres tipos:

- **Colaboración:** el crecimiento de la primera entidad de calidad implica el crecimiento de la segunda entidad de calidad.
- **Daño:** el crecimiento de la primera entidad de calidad implica la disminución de la segunda entidad de calidad.
- **Dependencia:** algunos valores del primer atributo de calidad requieren que el segundo cumpla ciertas condiciones.

**Paso 6. Determinación de métricas para los atributos**

No solo es necesario identificar los atributos sino que también lo es la selección de métricas para todos los atributos. Para esta tarea, podemos usar la teoría general de métricas.

#### 4. Modelo de calidad de un almacén de datos

En esta sección, explicamos como aplicar el método descrito en la sección anterior para desarrollar un modelo de calidad para almacenes de datos. Hemos desarrollado cada paso propuesto, e incluso hemos relacionado nuestras métricas definidas previamente con el modelo de calidad.

##### **Paso 0. Definición del dominio**

En este paso definimos el dominio para el que queremos construir un modelo de calidad, en este caso el esquema conceptual de un almacén de datos. A continuación, esbozamos una aproximación basada en UML (*Unified Modeling Language*) para el modelado conceptual de almacenes de datos. Esta aproximación se ha especificado por medio de un *profile* UML<sup>1</sup> que contiene los estereotipos necesarios para llevar a cabo el modelo conceptual del almacén de datos

<sup>1</sup> Un *profile* es un conjunto de mejoras cuya finalidad es la extensión de un tipo de diagrama UML existente para posibilitar un uso diferente. Estas mejoras se especifican por medio de mecanismos de extensibilidad como estereotipos, propiedades y restricciones.

de forma satisfactoria [15], mediante la representación de los principales conceptos multidimensionales a nivel conceptual (ver Tabla 1). Concretamente, las propiedades estructurales del modelado multidimensional se representan por medio de un diagrama de clases UML en el que la información se organiza de forma clara en hechos y dimensiones. Estos hechos y dimensiones se representan respectivamente por clases *Fact* y *Dimension*. Las clases *Fact* se definen como una composición de *n* clases *Dimension* en una agregación compartida (*shared aggregation*). La cardinalidad mínima del rol de la clase *Dimension* es 1 para indicar que cada hecho debe relacionarse siempre con cada una de las dimensiones. Un hecho está compuesto de medidas (o atributos de hecho). Estos se representan como atributos con el estereotipo *FactAttribute*. Por defecto, todas las medidas en la clase *Fact* se consideran aditivas. Para las medidas no-aditivas, las reglas de aditividad se definen como restricciones y se incluyen en la clase *Fact* correspondiente. Además, las medidas derivadas (indicadas por /) y sus reglas de derivación (indicadas por /) pueden representarse explícitamente por medio de valores etiquetados de un *FactAttribute*.

Nuestra aproximación también permite la definición de dimensiones degeneradas, con el fin de representar otras características del hecho además de las propias medidas. Estas dimensiones degeneradas se representan con atributos de la clase *Fact* estereotipados como *DegenerateDimension*.

Las relaciones muchos-a-muchos entre un hecho y una determinada dimensión se especifican mediante la cardinalidad 1..n en el rol de la clase *Dimension* correspondiente. En este caso, si se necesitara describir atributos específicos que representen características adicionales para cada combinación de instancias de esta relación, se usa una clase de asociación estereotipada como *DegenerateFact*. Esta clase representa hechos degenerados que pueden contener atributos estereotipados como *FactAttribute* o *DegenerateDimension*.

Con respecto a las dimensiones, cada nivel de una jerarquía de clasificación se especifica mediante una clase *Base*. Cada clase *Base* puede contener varios atributos de dimensión (estereotipo *DimensionAttribute*), un atributo identificador (estereotipo *OID*) y también debe contener un atributo descriptor (estereotipo *Descriptor*). Una

asociación (representada por un estereotipo llamado *Rolls-UpTo*) entre clases *Base* permite especificar la relación entre dos niveles de una jerarquía de clasificación. El único prerrequisito es que estas clases deben definir un Grafo Acíclico Dirigido (DAG) enraizado en la clase *Dimension* (la restricción del DAG se define en el estereotipo *Dimension*). La estructura del DAG puede representar caminos de jerarquía alternativos y múltiples. Una clase *Dimension* contiene un único primer nivel de jerarquía llamado nivel terminal de dimensión. Un camino de agregación es una subsecuencia de niveles de dimensión, que empieza en este nivel terminal (nivel de detalle menor) y finaliza en un nivel implícito (no representado gráficamente) que representan todos los niveles de dimensión.

Usamos roles para representar la forma en que dos clases *Base* se ven una a la otra en la asociación *Rolls-UpTo*: el rol R representa la dirección en la que la jerarquía se agrega mientras que el rol D representa la dirección en la que la jerarquía se desagrega. Además, usamos los roles para detectar y evitar ciclos en una jerarquía de clasificación, ayudando a conseguir la condición DAG.

Debido a la flexibilidad de UML, también podemos considerar jerarquías no-estrictas (un objeto de una jerarquía de un nivel inferior pertenece a más de objeto de nivel superior) y jerarquías completas (todos los miembros pertenecen un objeto de clase superior y ese objeto solo consta de esos miembros). Estas características se especifican, respectivamente, mediante la cardinalidad de los roles de las asociaciones y la definición del estereotipo *Completeness* en la asociación entre clases *Base*. Por último, la categorización de las dimensiones se considera por medio de las relaciones de generalización/especialización de UML.

Nuestro *profile* se define formalmente mediante el uso de OCL (*Object Constraint Language*) para expresar reglas bien formadas sobre los nuevos elementos definidos (ver Tabla 1), y así evitar su uso arbitrario. Remitimos al lector a [15] para una explicación más amplia de este *profile* y sus correspondientes restricciones OCL.

Estereotipo	Notación
<i>Fact</i>	
<i>Dimension</i>	
<i>Base</i>	
<i>DegenerateFact</i>	
<i>FactAttribute</i>	
<i>DegenerateDimension</i>	
<i>DimensionAttribute</i>	
<i>OID</i>	
<i>Descriptor</i>	
<i>Rolls-UpTo</i>	
<i>Completeness</i>	

Tabla 1. Principales estereotipos de nuestro *profile* para modelado multidimensional de almacenes de datos.

#### **Paso 1. Determinación de las subcaracterísticas de calidad**

En este paso las características de calidad se descomponen en subcaracterísticas siguiendo el estándar ISO 9126 y en las características del dominio con las que vamos a trabajar. El estándar ISO 9126 propone la descomposición mostrada en la Tabla 2

Características	Subcaracterísticas
<i>Funcionalidad (Functionality)</i>	Idoneidad (Suitability) Exactitud (Accuracy) Interoperabilidad (Interoperability) Seguridad (Security)
<i>Fiabilidad (Reliability)</i>	Madurez (Maturity) Tolerancia a fallos (Fault Tolerance) Recuperabilidad (Recoverability)
<i>Usabilidad (Usability)</i>	Entendibilidad (Understandability) Aprendizabilidad (Learnability) Operabilidad (Operability) Atractividad (Attractiveness)
<i>Eficiencia (Efficiency)</i>	Comportamiento Temporal (Time behaviour) Utilización de recursos (Resource utilization)
<i>Mantenibilidad (Maintainability)</i>	Análisis (Analyzability) Cambiability (Changeability) Estabilidad (Stability) Facilidad de Prueba (Testability)
<i>Portabilidad (Portability)</i>	Adaptabilidad (Adaptability) Instalabilidad (Installability) Coexistencia (Coexistence) Facilidad de reemplazo (Replaceability)

Tabla 2. Características y subcaracterísticas de ISO/IEC 9126-1

En nuestro dominio, los modelos conceptuales de los almacenes de datos, creemos firmemente que algunas de las dimensiones del modelo ISO 9126 no son aplicables, ya que la calidad de un esquema conceptual de un almacén de datos está relacionada con la corrección, completitud, comprensión, seguridad y estabilidad del esquema. De todas estas dimensiones incluidas en la norma ISO 9126, hemos eliminado aquellas que no están directamente relacionadas con el modelado conceptual, sino más bien con aspectos de la implementación final del almacén de datos. En concreto, hemos eliminado la *eficiencia*, *fiabilidad* y *portabilidad* porque se refieren más a aspectos de comportamiento, recursos temporales del sistema del almacén de datos, tolerancia a fallos, recuperabilidad del sistema, etc. En la Tabla 3 podemos encontrar las subcaracterísticas que se han tenido en cuenta después de eliminar las que no son aplicables a nuestro dominio.

Características (Functionality)	Sub-características (Suitability)
Exactitud (Accuracy)	Idoneidad (Suitability)
Seguridad (Security)	Exactitud (Accuracy)
Usabilidad (Usability)	Seguridad (Security)
Mantenibilidad (Maintainability)	Entendibilidad (Understandability)
	Aprendizabilidad (Learnability)
	Analizabilidad (Analyzability)
	Cambiability (Changeability)
	Estabilidad (Stability)

Tabla 3. Conjunto final de subcaracterísticas de calidad

**Paso 2. Definición de una jerarquía de subcaracterísticas**

En nuestro contexto (la calidad de los modelos conceptuales de un almacén de datos), no hay necesidad de descomponer las subcaracterísticas evaluadas en el paso anterior en otras más pequeñas.

**Paso 3. Descomposición de las subcaracterísticas en atributos**

Como hemos afirmado previamente, las subcaracterísticas de calidad proporcionan una vista comprensible del modelo de calidad pero son demasiado abstractas y difíciles de medir. En este paso, los conceptos abstractos se descomponen en atributos. Para cada característica y subcaracterística relevante, proporcionamos un conjunto de atributos relacionados con esa característica (ver Tabla 4)

Características (Functionality)	Subcaracterísticas (Suitability)	Atributo
Exactitud (Accuracy)	Idoneidad (Suitability)	Adecuación de los datos modelados a los requisitos
		Complejidad
		Especificación Correcta
Seguridad (Security)	Seguridad (Security)	Adecuación a los requisitos
		Correctitud Semántica

Usabilidad (Usability)	Entendibilidad (Understandability)	Atributo
Usabilidad (Usability)	Entendibilidad (Understandability)	Facilidad de entendimiento
		Complejidad del esquema
		Tamaño del esquema
Aprendizabilidad (Learnability)	Aprendizabilidad (Learnability)	Facilidad de entendimiento
		Complejidad del esquema
		Tamaño del esquema
Analizabilidad (Analyzability)	Analizabilidad (Analyzability)	Facilidad de entendimiento
		Complejidad del esquema
		Profundidad del esquema

Mantenibilidad (Maintainability)	Analizabilidad (Analyzability)	Atributo
Mantenibilidad (Maintainability)	Analizabilidad (Analyzability)	Facilidad de entendimiento
		Complejidad del esquema
		Tamaño del esquema
Cambiability (Changeability)	Cambiability (Changeability)	Facilidad de entendimiento
		Complejidad del esquema
		Profundidad del esquema
Estabilidad (Stability)	Estabilidad (Stability)	Complejidad del esquema
		Acoplamiento
		Normalización

Tabla 4. Atributos relacionados con cada subcaracterística de la calidad

**Paso 4. Descomposición de los atributos derivados en atributos básicos**

Después del paso 3 podemos encontrar que algunos atributos son demasiado abstractos para ser medidos fácilmente, debiéndose descomponer en otros más concretos. Sin embargo, los atributos presentados en la Tabla 4 son suficientemente concretos y por lo tanto es innecesario descomponerlos.

**Paso 5. Establecer relaciones entre las entidades de calidad**

En este paso establecemos las relaciones entre las entidades de calidad, haciendo que el modelo sea más exhaustivo y sus implicaciones

más claras. La Tabla 5 muestra las relaciones entre los atributos de calidad. En esta tabla el término 'COL' corresponde con *Colaboración*, 'DEP' con *Dependencia* y 'DMG' quiere decir *Daño*.

Características (Functionality)	Sub-características (Suitability)	Atributo	Métrica
Exactitud (Accuracy)	Idoneidad (Suitability)	Adecuación a los requisitos	Número de requisitos de información redundantes
			Número de requisitos de información implementados en el modelo
Seguridad (Security)	Seguridad (Security)	Correctitud Semántica	Número de especificaciones erróneas
			Número de especificaciones redundantes
Usabilidad (Usability)	Entendibilidad (Understandability)	Facilidad de entendimiento	Número de requisitos de información implementados en el modelo
			Número de atributos multidimensionales que faltan
Aprendizabilidad (Learnability)	Aprendizabilidad (Learnability)	Complejidad del esquema	Número de elementos multidimensionales que faltan representados en los requisitos
			Número de jerarquías de dimensión que faltan
Analizabilidad (Analyzability)	Analizabilidad (Analyzability)	Profundidad del esquema	Número de requisitos de información no implementados en el modelo
			Número de requisitos de información implementados en el modelo
Mantenibilidad (Maintainability)	Analizabilidad (Analyzability)	Complejidad del esquema	Número de especificaciones redundantes
			Número de requisitos de información implementados en el modelo
Cambiability (Changeability)	Cambiability (Changeability)	Acoplamiento	Número de requisitos de información implementados en el modelo
			Número de atributos multidimensionales que faltan representados en los requisitos
Estabilidad (Stability)	Estabilidad (Stability)	Normalización	Número de requisitos de información implementados en el modelo
			Número de atributos multidimensionales que faltan representados en los requisitos

Tabla 5. Relaciones entre los atributos de calidad

Por ejemplo en la Tabla 5, podemos ver que la *Modularidad* colabora con la *Facilidad de entendimiento* porque si el sistema es modular es probable que sea más fácil de entender. Esta tabla es un resumen de todas las relaciones entre los atributos de calidad.

**Paso 6. Determinación de métricas para los atributos**

Después de la identificación de los atributos, debemos asignar métricas a cada atributo. La Tabla 6 muestra las métricas propuestas para cada atributo. Parte de estas métricas se basan en nuestros trabajos previos presentados en [16, 17], por lo que ahora, estas métricas quedan englobadas en un modelo de calidad basado en el estándar ISO 9126.

Características (Functionality)	Sub-características (Suitability)	Atributo	Métrica
Exactitud (Accuracy)	Idoneidad (Suitability)	Adecuación a los requisitos	Número de requisitos de información redundantes
			Número de requisitos de información implementados en el modelo
Seguridad (Security)	Seguridad (Security)	Correctitud Semántica	Número de especificaciones erróneas
			Número de especificaciones redundantes
Usabilidad (Usability)	Entendibilidad (Understandability)	Facilidad de entendimiento	Número de requisitos de información implementados en el modelo
			Número de atributos multidimensionales que faltan
Aprendizabilidad (Learnability)	Aprendizabilidad (Learnability)	Complejidad del esquema	Número de elementos multidimensionales que faltan representados en los requisitos
			Número de jerarquías de dimensión que faltan
Analizabilidad (Analyzability)	Analizabilidad (Analyzability)	Profundidad del esquema	Número de requisitos de información no implementados en el modelo
			Número de requisitos de información implementados en el modelo
Mantenibilidad (Maintainability)	Analizabilidad (Analyzability)	Complejidad del esquema	Número de especificaciones redundantes
			Número de requisitos de información implementados en el modelo
Cambiability (Changeability)	Cambiability (Changeability)	Acoplamiento	Número de requisitos de información implementados en el modelo
			Número de atributos multidimensionales que faltan representados en los requisitos
Estabilidad (Stability)	Estabilidad (Stability)	Normalización	Número de requisitos de información implementados en el modelo
			Número de atributos multidimensionales que faltan representados en los requisitos

Usabilidad (Usability)	Seguridad (Security)	Presencia de restricciones de seguridad	Número de restricciones de seguridad	
		Ratio de restricciones de seguridad	Ratio de restricciones de seguridad	
	Entendibilidad (Understandability)	Adecuación de las reglas de seguridad	Complejidad de reglas de seguridad	Número de reglas de seguridad
			Facilidad de entendimiento	Complejidad de reglas de seguridad
		Complejidad del esquema	Número de clases de hecho, dimensión y base	Número de clases de hecho, dimensión y base
			Ratio de relaciones en las jerarquías de dimensión	Ratio de relaciones en las jerarquías de dimensión
		Tamaño del esquema	Adecuación de los nombres	Adecuación de los nombres
			Número de relaciones en las jerarquías de dimensión	Número de relaciones en las jerarquías de dimensión
		Profundidad del esquema	Número de relaciones en las jerarquías de dimensión y base	Número de relaciones en las jerarquías de dimensión y base
			Longitud del esquema	Longitud del esquema
Facilidad de entendimiento		Profundidad máxima de las jerarquías de dimensión	Profundidad máxima de las jerarquías de dimensión	
		Número de relaciones en las jerarquías de dimensión y base	Número de relaciones en las jerarquías de dimensión y base	
Mantenibilidad (Maintainability)	Aprendizabilidad (Learnability)	Ratio de relaciones en las jerarquías de dimensión	Ratio de relaciones en las jerarquías de dimensión	
		Adecuación de los nombres	Adecuación de los nombres	
	Complejidad del esquema	Número de relaciones en las jerarquías de dimensión	Número de relaciones en las jerarquías de dimensión	
		Ratio de relaciones en las jerarquías de dimensión y base	Ratio de relaciones en las jerarquías de dimensión y base	
	Tamaño del esquema	Número de relaciones en las jerarquías de dimensión	Número de relaciones en las jerarquías de dimensión	
		Profundidad del esquema	Profundidad del esquema	
	Facilidad de entendimiento	Número de relaciones en las jerarquías de dimensión y base	Número de relaciones en las jerarquías de dimensión y base	
		Ratio de relaciones en las jerarquías de dimensión	Ratio de relaciones en las jerarquías de dimensión	
	Complejidad del esquema	Adecuación de los nombres	Adecuación de los nombres	
		Número de relaciones en las jerarquías de dimensión	Número de relaciones en las jerarquías de dimensión	
Tamaño del esquema	Número de relaciones en las jerarquías de dimensión y base	Número de relaciones en las jerarquías de dimensión y base		
	Profundidad del esquema	Profundidad del esquema		
Cambabilidad (Changeability)	Acompilamiento	Profundidad máxima de las jerarquías de dimensión	Profundidad máxima de las jerarquías de dimensión	
		Cohesión	Cohesión del esquema	
	Normalización	Cohesión del esquema	Cohesión del esquema	
		Adecuación a las normas	Adecuación a las normas	
	Modularidad	Número de paquetes de dimensión y hecho	Número de paquetes de dimensión y hecho	
		Ratio de paquetes de dimensión y hecho	Ratio de paquetes de dimensión y hecho	
	Facilidad de entendimiento	Número de clases de hecho, dimensión y base	Número de clases de hecho, dimensión y base	
		Ratio de relaciones en las jerarquías de dimensión	Ratio de relaciones en las jerarquías de dimensión	
	Complejidad del esquema	Adecuación de los nombres	Adecuación de los nombres	
		Número de relaciones en las jerarquías de dimensión	Número de relaciones en las jerarquías de dimensión	
Acompilamiento	Ratio de relaciones en las jerarquías de dimensión	Ratio de relaciones en las jerarquías de dimensión		
	Cohesión del esquema	Cohesión del esquema		
Estabilidad (Stability)	Normalización	Adecuación a las normas	Adecuación a las normas	
		Número de paquetes de dimensión y hecho	Número de paquetes de dimensión y hecho	
Modularidad	Cohesión del esquema	Ratio de paquetes de dimensión y hecho	Ratio de paquetes de dimensión y hecho	
		Cohesión del esquema	Cohesión del esquema	

Table 6. Métricas asociadas a cada característica de calidad.

## 5. Conclusiones y trabajos futuros

Hoy en día, los almacenes de datos son un producto software muy relevante en sistemas de apoyo a la decisión. Debido a las características de los entornos donde se usan los almacenes de datos, asegurar la calidad de estos sistemas es una cuestión crucial. Cuando se trata de la calidad de un almacén de datos, es importante evaluar la calidad tanto de los datos como de los esquemas, ya que la calidad del modelo que soporta los datos influye en la calidad de los mismos y también en la del sistema completo.

Para evaluar la calidad del esquema de un almacén de datos es necesario definir un modelo de calidad, con el fin de ayudar a los diseñadores a asegurar la calidad en el sistema final cuando éste se desarrolla de una forma objetiva y sistemática.

En este artículo hemos revisado un método para la creación de un modelo de calidad basado en el estándar ISO 9126 y hemos aplicado sus seis pasos para crear una primera propuesta de un modelo de calidad para esquemas conceptuales de almacenes de datos.

Como resultado de este método hemos obtenido un conjunto de características de calidad importantes que se dividen en subcaracterísticas y atributos de calidad. Como paso final hemos propuesto un conjunto inicial de métricas para evaluar esos atributos de calidad y los hemos relacionado con nuestras métricas validadas y definidas previamente.

El siguiente paso en nuestra propuesta de un modelo de calidad para almacenes de datos tiene que ver con su validación. Planeamos empezar con la validación de las métricas propuestas para evaluar la utilidad y la adecuación. Después, creemos que deberíamos intentar validar el modelo de calidad completo en un entorno industrial.

## Agradecimientos

Este trabajo está parcialmente financiado por el proyecto METASIGN (TIN2004-0779) financiado por el Ministerio de Educación y Ciencia, por el proyecto DADS (PBC-05-012-02) financiado por la Junta de Castilla-La Mancha y el proyecto CALIA financiado por la Universidad de Castilla-La Mancha. Jose-Norberto Mazón dispone de una beca FPU (AP2005-1360) del Ministerio de Educación y Ciencia.

## Referencias

- [1] B. Shin, "An Exploratory Investigation of System Success Factors in Data Warehousing," *Journal of Association for Information Systems*, vol. 4, pp. 141-170, 2003.
- [2] T. Chenoweth, D. Schuff, and R. St. Louis, "A method for developing Dimensional data marts," *Communications of the ACM*, vol. 46, pp. 93-98, 2003.
- [3] P. Vassiliadis, "Gulliver in the land of data warehousing: practical experiences and observations of a researcher," presented at International Workshop on Design and Management of Data Warehouses (DMDW'2000), Stockholm (Sweden), 2000.
- [4] C. Stedman, "Warehousing Projects Hard to Finish," *Computerworld*, vol. 32, pp. 29, 1998.
- [5] N. Prat and S. S.-S. Cherfi, "Multidimensional Schemas Quality Assessment," presented at The 15th Conference on Advanced Information Systems Engineering (CAISE '03) Workshops, Klagenfurt/Velden (Austria), 2003.
- [6] ISO/IEC, "9126-1: Software Engineering - Product quality - Part 1: Quality model," 2001.
- [7] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, "The Capability Maturity Model for software," Software Engineering Institute, Carnegie Mellon University CMU/SEI-93-TR-024, February 2003 1993.
- [8] K. El Emam, J. N. Drouin, and W. Melo, "Spice: The Theory and Practice of Software Process Improvement and Capability Determination," IEEE Computer Society 1998.
- [9] IEEE, "IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology," 1998.
- [10] V. Basili and D. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, vol. 10, pp. 728-738, 1984.
- [11] V. Basili and H. Rombach, "The TAME project: towards improvement-oriented software

- environments," *IEEE Transactions on Software Engineering*, vol. 14(6), pp. 728-738, 1988.
- [12] M. Serrano, "Definition of a set of metrics for assuring data warehouse quality," University of Castilla - La Mancha (Spain), 2004.
- [13] M. Jarke, M. Lenzenini, Y. Vassiliou, and P. Vassiliadis, *Fundamentals of Data Warehouses*, second edition ed: Springer-Verlag, 2002.
- [14] X. Franch and J. P. Carvallo, "Using Quality Models in Software Package Selection," *IEEE Software*, vol. 20, pp. 34-41, 2003.
- [15] S. Luján-Mora, J. Trujillo, and I.-Y. Song, "A UML profile for multidimensional modeling in data warehouses," *Data & Knowledge Engineering (DKE)*, vol. 59, pp. 725-769, 2006.
- [16] M. Serrano, C. Calero, J. Trujillo, S. Luján-Mora, M. Piattini. Empirical Validation of Metrics for Conceptual Models of Data Warehouses. CAISE 2004, LNCS 3084, pp. 506-520.
- [17] G. Berenguer, R. Romero, J. Trujillo, M. Serrano, M. Piattini. A Set of Quality Indicators and Their Corresponding Metrics for Conceptual Models of Data Warehouses. DaWak 2005, LNCS 3589, pp. 95-104.