

IV

**SIMPOSIO INTERNACIONAL
DE SISTEMAS DE INFORMACIÓN
E INGENIERÍA DE SOFTWARE EN
LA SOCIEDAD DEL CONOCIMIENTO**

SI SOFT 2006

Entidades Organizadoras



UNIVERSIDAD DISTRITAL FRANCISCO
JOSÉ DE CALDAS (COLOMBIA)



UNIVERSIDAD PONTIFICIA DE
SALAMANCA (ESPAÑA)



FUNDACIÓN PABLO VI
(ESPAÑA)

LIBRO DE ACTAS VOLUMEN I

Agosto 23 al 25 de 2006
Cartagena de Indias - Colombia

Editado por :

Luis Joyanes Aguilar
Universidad Pontificia de Salamanca

www.sissoftw.com

ISBN: 84-690-0258-9

ENTIDADES COORGANIZADORAS:



Colciencias(Colombia)



Universidad de Cartagena (Colombia)



Asociación Colombiana de Informatica(Colombia)



Instituto Tecnológico de las Américas(Repu. Dominicana)



Pontificia Universidad del Peru(Peru)

ENTIDADES COLABORADORAS:



Rama Estudiantil IEEE-UD (Colombia)



Universidad de Oviedo (España)



Centro de Tecnología Avanzada(Mexico)



Universidad Católica de Colombia(Colombia)



Universidad Privada Antenor de Orrego(Peru)



Centro de Supercomputación de Barcelona(España)



Tompkins Cortland Community College(EEUU)



Tecnológico de Monterrey(Mexico)



Universidad Autónoma de Yucatán (Mexico)



Universidad de las Palmas de Gran Canaria(España)



Universidad Apec (Rep Dominicana)



Escuela Superior Politécnica del litoral(Ecuador)



Universidad Autónoma de Occidente(Colombia)



Universidad Católica de Montevideo (Uruguay)



Universidad Nacional de Nordeste(Argentina)

ENTIDADES PATROCINADORAS



Avianca (Colombia)



Banco Occidente (Colombia)



Famisanar (Colombia)



Universia



Trotamundos (Colombia)



Empresa de Telefonos de Bogota(Colombia)



Microsoft



Fundación de Educación Superior San José(Colombia)

**IV SIMPOSIO INTERNACIONAL DE
SISTEMAS DE INFORMACIÓN E
INGENIERÍA DE SOFTWARE EN LA
SOCIEDAD DEL CONOCIMIENTO**

SISOFT2006

VOLUMEN I

**Cartagena de Indias, Colombia
23, 24 y 25 de Agosto de 2005**

www.sisoftw.com

**Editor
Luís Joyanes Aguilar**

**UNIVERSIDAD PONTIFICIA DE SALAMANCA
(CAMPUS MADRID)**

FUNDACIÓN PABLO VI

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS

PONTIFICIA UNIVERSIDAD CATÓLICA DE PERÚ

IV SIMPOSIO INTERNACIONAL DE SISTEMAS DE INFORMACIÓN E INGENIERIA DE SOFTWARE EN LA SOCIEDAD DEL CONOCIMIENTO

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

DERECHOS RESERVADOS © 2006

Universidad Pontifica de Salamanca (España)
Fundación Pablo VI (España)
Universidad Distrital Francisco José de Caldas (Colombia)
Instituto Tecnológico de las Américas (ITLA, República Dominicana)
Pontificia Universidad Católica del Perú (Perú)

ISBN: 84-690-0258-9

Editor:

Luís Joyanes Aguilar (FI/UPSAM, España)

Adjuntos al Editor:

Víctor Hugo Medina García (FI/UDFJC, Colombia)
Javier Parra Fuente (FI/UPSAM, España)
Oscar Sanjuán Martínez (FI/UPSAM, España)
Daniel Zapico Palacio (FI/UPSAM, España)
Juan Manuel Lombardo Enríquez (FI/UPSAM, España)

Edita:

Universidad Pontifica de Salamanca campus de Madrid (España)
Fundación Paulo VI, Madrid (España)
Universidad Distrital Francisco José de Caldas, Bogotá (Colombia)

IMPRESO EN BOGOTÁ (COLOMBIA)

DIRECCIÓN DEL SIMPOSIO

- **Dr. Luis Joyanes Aguilar** (Presidente) (UPSAM, España)
- **Dr. Víctor Hugo Medina García** (Director Ejecutivo) (U. Distrital F.J.C., Colombia)

COMITÉ DE PROGRAMA PERMANENTE (STEERING COMMITTEE)

- **Dr. Luis Joyanes Aguilar** (Presidente) (UPSAM, España)
- **Dr. Victor Hugo Medina García** (Director Ejecutivo) (UDFJC, Colombia)
- **Msc. José Armando Tabares** (ITLA, Rep. Dominicana)
- **Dr. Juan Manuel Cueva** (U. Oviedo, España)
- **Dr. Maynard Kong** (PUCP, Perú)
- **Dr. Javier Parra Fuente** (UPSAM, España)

COMITÉ DE HONOR

- **Excmo. y Sr. Magnífico D. Marceliano Arranz Rodrigo** (Rector UPSA -España)
- **Excmo. y Rvdo. Sr. D. Fernando Sebastián Aguilar** (Presidente Fundación Pablo VI - España)
- **Excmo. y Rvdo. Sr. D. Ángel Berna Quintana** (Director General Fundación Pablo VI - España)
- **Excmo. Sr. Magnífico D. Ricardo García Duarte** (Rector UDFJC - Colombia)
- **Excmo. Sr. Magnífico D. Rafael Peña Suesca** (Decano Facultad de Ingeniería UDFJC - Colombia)
- **Excmo. Sr. Magnífico D. Germán Sierra Anaya** (Rector Universidad de Cartagena - Colombia)
- **Don Luis Eduardo Garzón** (Alcalde Mayor de Bogotá - Colombia)

COMITÉ DE ORGANIZACIÓN (EUROPEO)

- **Dr. Juan Manuel Lombardo** (Coordinador General) (UPSAM, España)
- **Dr. Javier Parra** (Org. Técnica) (UPSAM, España)
- **Dr. Oscar Sanjuán Martínez** (Org. Técnica) (UPSAM, España)
- **DEA. Daniel Zapico Palacio** (Org. Técnica) (UPSAM, España)
- **Lic. Teresa Díez** (UPSAM, España)
- **Raquel Ureña** (UPSAM, España)

COMITÉ DE ORGANIZACIÓN (COLOMBIANO)

- **Dr. Víctor Hugo Medina García** (Coordinador) (UDFJC)
- **Dr. Jairo Torres Acosta** (UDFJC)
- **Dr. Nelson Pérez Castillo** (UDFJC)
- **Ing. Msc. Anselmo Vega Vega** (UDFJC)
- **Ing. Msc. Edmundo Vega Osorio** (UDFJC)
- **Ing. Giovany Tarazona Bermúdez** (UDFJC)
- **Ing. Beatriz Jaramillo** (UDFJC)
- **Ing. Alexis Ortiz Morales** (UDFJC)
- **Ing. Carlos Andrés Torres** (UDFJC)
- **Ing. Carlos Eduardo Vargas** (UDFJC)
- **Ing. Jesús Guzmán** (UDFJC)
- **Ing. Diana Ahumada** (UDFJC)

COMITÉ DE CO-ORGANIZACIÓN (COLOMBIANO)

- **Ing. Msc. Alfonso Pérez Gama** (ACCIO)
- **Ing. Msc. Pedro Guardela** (U. Cartagena)
- **Ing. Msc. Julio César Rodríguez Ribón** (U. Cartagena)
- **Ing. Msc. David Antonio Franco Borré** (U. Cartagena)
- **Ing. Msc. Miguel Ángel García Bolaños** (U. Cartagena)
- **Ing. Msc. Isaac Zúñiga Silgado** (U. Tecnológica de Bolívar)
- **Ing. Msc. Juan Antonio Contreras** (Corporación Universitaria Rafael Núñez)

COMITÉ DE ORGANIZACIÓN (DOMINICANO)

- **Ing. José Armando Tabares** (Director Ejecutivo, ITLA)
- **Lic. Quinta Ana Pérez** (Secretaría técnica, ITLA)

COMITÉ DE ORGANIZACIÓN (PERUANO)

- **José Antonio Pow-Sang** (PUCP)
- **Ing. Abraham Dávila** (PUCP)

COMITÉ INTERNACIONAL DE APOYO

- **MSc. Oscar Mendoza Macías** (ESPOL, Ecuador)
- **MSc. Jorge Huayahuaca** (TC3, EE.UU.)
- **MSc. Ernesto Ocampo** (U. Católica de Uruguay)
- **MSc. Juan Estanis** (Argentina)
- **Dr. Arturo Fernández** (TEC de Monterrey, México)
- **Ing. Jorge Torres** (TEC de Monterrey, México)
- **MSc. Inmaculada Madero** (UNAPEC, República Dominicana)
- **MSc. Miguel Cid** (INTEC, República Dominicana)
- **MSc. Luz Mayela Ramirez** (U. Católica de Colombia)
- **MSc. Jesús Cardona** (U.A. de Cali, Colombia)
- **MSc. Elmer González** (UPAO, Perú)
- **Ing. José Fernando Baquero Herrero** (CIATEQ, México)

COMITÉ DE PROGRAMA

- **Dr. Luís Joyanes Aguilar** (Presidente) (UPSAM, España)
- **Dr. Álvaro Suárez** (ULPGC, España)
- **Dr. Angel Egido** (U. Catholique d'Angers, Francia)
- **Dr. Ángel Losada** (UPSA, España)
- **Dr. Arturo Ribagorda** (UC3M, España)
- **Dr. Darío Álvarez** (UNIOVI, España)
- **Dr. David Olivieri** (UV, España)
- **MSc. Esteban Bolondi** (U. Católica de Colombia)
- **Dr. Francisco Rubio** (ULPGC, España)
- **Dr. Gustavo Rossi** (U de la Plata, Argentina)
- **Dr. Javier Bustamante** (UCM, España)
- **Dr. Javier Segovia** (UPM, España)
- **Dr. Javier Nó** (UPSA, España)
- **Dr. Jean Pierre Boutinet** (U. Catholique d'Angers, Francia)
- **Dr. Jesús de la Cruz Escotto**, (USAC, Guatemala)
- **Dr. José R. García-Bermejo** (USA, España)
- **Dr. José Ramón Pin** (IESE- Madrid, España)
- **Dr. Juan Manuel Corchado** (USA, España)
- **Dr. L. Alfonso Ureña López** (UJAEN, España)
- **Dr. Luis Rodríguez Baena** (UPSAM, España)
- **Dr. Manuel Maceiras** (CISTEC, España)
- **Dra. Matilde Fernández** (UPSAM, España)

- **Dra. Mercedes Caridad** (UC3M, España)
- **Dr. Miguel Katrib** (U. de la Habana, Cuba)
- **Dr. Nelson Pérez Castillo** (UDFJC, Colombia)
- **Dr. Sebastián Dormido** (UNED, España)
- **Dr. Víctor Martín García** (UPSAM, España)
- **Dr. Santos Gracia** (Fundación Universitaria Iberoamericana)
- **Ing. Mag. Abraham Dávila** (PUCP, Perú)
- **MsC. Ernesto Ocampo Edye** (UCU, Uruguay)
- **Ing. Carlos M. Fernández** (Aenor, UPSAM, España)
- **Ing. David La Red Martínez** (UNN, Argentina)
- **Dr. Mateo Valero** (BSC, España)
- **Dr. José Antonio Moreira** (UC3M, España)
- **Dr. Mario Piattini** (UCLM, España)
- **Dr. Fernando Martín** (UM, España)
- **Dra. Teresa Peña** (UB, España)
- **Ing. Fernando Davara** (UPSA, España)
- **Ing. Miguel Rego** (ASIMELEC, España)
- **Dr. Víctor Hugo Medina** (UDFJC, Colombia)
- **DEA. Jorge Torres** (TEC, México)
- **Dr. Vidal Alonso Secades** (UPSA, España)
- **Dr. Alfonso López Rivero** (UPSA, España)
- **Ing. Fernando Curi** (UADY, México)
- **Dr. Manuel Pérez Cota** (U.Vigo, España).
- **Dr. Javier Ariza** (U. Jaén, España)

CONTENIDO

PROLOGO

AUDITORÍA..... 12

Defining, Performing and Maintaining Software Measurement Programs: State of the Art..... 13

BIOINFORMÁTICA 24

Redes neuronales para la predicción de proteínas..... 25

Clasificación Automática Sensible al costo para la Detección de Neuropatías Periféricas Focales..... 39

COMERCIO ELECTRÓNICO..... 47

Modelo Arquitectónico Neutral Para la Interoperabilidad de Plataformas de Gestión del Aprendizaje 48

CERTILOC: un mecanismo seguro para m-Marketing y Comercio electrónico basados en servicios de localización 58

EDUCACIÓN 69

E-Learning y Espacios Colaborativos..... 70

Estudio de viabilidad de la aplicación de Sistemas de Recomendación a entornos de e-learning 77

Las Tecnologías de la Información y de las Comunicaciones y los modelos integrados en la Educación – Combatir el fracaso en Enseñanza y tornarlos más eficaces en su aplicación 85

Metafora de aula de clase como ambiente virtual en el proceso enseñanza-aprendizaje 99

Un Proceso Ágil para el Mejoramiento de Procesos de Desarrollo de Software para PYMES – Agile SPI – Process. 108

INDUSTRIA DEL SOFTWARE 116

Utilización de las tecnologías de Información en el aula 117

Pruebas en Programación Extrema 126

Migra-T: Una Herramienta para Migrar Procedimientos Almacenados sobre Múltiples Motores de Base de Datos Relacionales Comerciales 136

Calidad de Productos de Software: Un estado del arte de la medición 145

Incorporación de medidas en el modelo de procesos para la industria de software MoProSoft..... 153

Prototipo de Software para el preprocesamiento de datos “UD-Clear” 167

Un enfoque pragmático para aplicar tecnologías de análisis estático a la calidad del software 185

INGENIERÍA DEL SOFTWARE..... 194

Business Activity Monitoring y Business Rules para el Manejo de Excepciones en las Políticas en un sistema de Gestión de Procesos de Negocios. Estado del -Arte 195

Hacia un Nuevo Paradigma de Acceso: El Acceso Local Comunitario..... 204

Flexible Querying of XML Knowledge Base through the MIEL Language..... 212

La Reutilización del Conocimiento en Ambientes de Desarrollo de Software 221

Sistema para el etiquetado de discursos orales aplicado al nuevo sistema acusatorio penal..... 230

Integración de las actividades de Staff a WS-BPEL. Estado del Arte..... 241

Evolución de las metodologías de desarrollo de software a las orientadas a agentes 249

La Granja Integral, una Aplicación de Empresa Basada en Conocimientos 262

Generación de Mapas de Conocimiento Organizacionales a partir de instancias de procesos basadas en el Lenguaje de Ejecución de Procesos de Negocios (BPEL).....	272
INGENIERÍA MULTIMEDIA Y REALIDAD VIRTUAL	281
Desarrollo de un sistema de captura de movimiento para interacción en entornos virtuales	282
Diseño, Construcción e Implementación de una Consola de Efectos Digitales para Guitarra Eléctrica (Contemporánea) Mediante Software en Tiempo Real.....	289
INTELIGENCIA DE NEGOCIOS Y GESTIÓN DEL CONOCIMIENTO	298
Redes Neuronales Artificiales Una Aplicación A La Tasa De Cambio Nominal	299
Nuevas formas en la captura de datos para la Gestion Documental.....	316
El conocimiento tácito en la mejoría continúa de los proyectos	323
Captura de Necesidades en la Dirección y Gestión del Capital Intelectual.....	331
LENGUAJES DE PROGRAMACIÓN Y NUEVOS PARADIGMAS DE LA INGENIERÍA DEL SOFTWARE.....	339
Extensión de los lenguajes orientados a objetos con mecanismos de recuperabilidad dirigidos por anotaciones.....	341
Hacia un Marco de Trabajo para la Definición de Procesos de Desarrollo de Software; Framework-PDS.....	353
Una recomendación para la implantación de SOA (Service Oriented Architecture) en un contexto de Negocio Bajo Demanda	365
RESPONSABILIDAD SOCIAL CORPORATIVA.....	373
La RSC como estrategia de negocio en las empresas tecnológicas.....	374
SEGURIDAD INFORMÁTICA Y PROTECCIÓN DE SISTEMAS.....	390
Diseño De Un Sistema De Seguridad De Informacion Con JCE Que Permita Asegurar El Tráfico De Información Entre Un SGBDR y Sus Clientes En Una Intranet Pequeña.	391
Metodología para la recolección y análisis de evidencia digital	399
SERVICIOS E INGENIERÍA WEB	410
Construcción de Servicios Web para el Sistema de Información Ambiental de Colombia.....	411
Servicio Web Semántico aplicado a los Modelos Digitales de Terreno.....	421
Modelo de gestión para procesos académico administrativos, en ambientes educativos flexibles, aplicando tecnologías de la Web semántica.	427
SISTEMAS DE INFORMACIÓN GEOGRÁFICA (GIS).....	435
Arquitectura distribuida para el descubrimiento de conocimiento y minería de datos geográficos en imágenes.....	436
SOCIEDAD DE LA INFORMACIÓN Y DEL CONOCIMIENTO	445
Una Aplicación a la Sociedad del Conocimiento: Metro Ethernet.....	446
Gestión del Conocimiento y Estructura Empresarial en una Organización Estudiantil Sin Animo de Lucro.....	457
Una Propuesta para la Evaluación de Grupos de Investigación, Desarrollo e Innovación mediante 3 variables: Motivación, Conocimientos y Gestión.....	464
Una revisión de las técnicas relacionadas con la Inteligencia Artificial y la Composición Musical Asistida por Ordenador (CAO)	472
Modelo de Gestión del Conocimiento en las Pymes Colombianas	484
Redes Ópticas: Una Aplicación para la Sociedad del Conocimiento	497

Auditoría de Conocimiento en una Pyme Colombiana - Un caso de estudio.....515

PRÓLOGO

Una breve reseña histórica de SISOFT

En Agosto de 2001, las universidades Pontificia de Salamanca (*campus* Madrid, España) y la Distrital Francisco José de Caldas (Bogotá, Colombia) junto con la Fundación Pablo VI (Madrid, España) organizaron el primer SISOFT en la ciudad de Bogotá. En el mes de Agosto de 2006 se cumplen cinco años, 3 ediciones realizadas y la 4ª edición que ahora celebramos. Bogotá, Lima, Santo Domingo y Cartagena de Indias son las cuatro ciudades donde se ha organizado y celebrado el Simposio de Ingeniería de Software y Sistemas de Información. En estos cinco años, la Ingeniería de Software, los Sistemas de Información y la ya madura Sociedad de la Información y la naciente Sociedad del Conocimiento, han evolucionado considerablemente en paralelo con las grandes innovaciones tecnológicas. SISOFT ha tratado de adaptarse a todos estos grandes cambios y prueba de ello son los dos volúmenes publicados del actual Libro de Actas.

Cuando en el año 2000, las universidades fundadoras decidieron poner en marcha un congreso internacional que pudiera servir de puente entre Europa y Latinoamérica, y en particular entre España y Colombia como países organizadores, en áreas de futuro en el mundo del software, se pensó que dada la dificultad y el gran reto que suponía su organización, deberíamos de organizarlo con espíritu de continuidad y con una periodicidad bianual. Así y tras el éxito del primer simposio, en Agosto del 2001, la Pontificia Universidad Católica de Perú en Lima, asumió el nuevo reto de organización del II Simposio que se celebró en Agosto de 2003, ya con una gran expansión en las zonas geográficas de Latinoamérica y Europa; en 2005, el Instituto Tecnológico de las Américas (ITLA) de Santo Domingo en la República Dominicana, asumió la organización de la tercera edición, donde ya su carácter internacional quedó totalmente asentado, con la incorporación de ponentes y conferenciantes de numerosos países americanos y europeos. Esta circunstancia llevó al Comité Fundador a pensar en el cambio de periodicidad y de este forma se ha pasado del carácter bianual al carácter anual, y la celebración en 2006 en la histórica, universitaria y hermosa ciudad de Cartagena de Indias, en el Caribe colombiano. El generoso apoyo de las universidades locales hizo que las actuales cinco entidades internacionales, que forman el Comité Organizador apostaran decididamente por una nueva organización que permitiera mostrar durante sus jornadas, los actuales cambios producidos en la media década pasada en las áreas citadas y en los temas fundamentales del simposio.

La Universidad Distrital Francisco José de Caldas con la Universidad Pontificia de Salamanca han asumido de nuevo el gran reto que suponía la periodicidad anual con un programa del Simposio que contemple los ámbitos científicos, académicos, profesionales, económicos junto con el estudio de las innovaciones tecnológicas que ya han impactado en la primera mitad de esta década, y sin duda ya están afectando y afectarán, al menos, a la segunda mitad, también, de esta década.

El Comité Fundacional desea agradecer el esfuerzo científico e investigador de la Universidad Distrital, las universidades locales de Cartagena, que han apoyado el Simposio, y en general de las numerosas universidades colombianas, restantes latinoamericanas y europeas que lo apoyan de una u otra forma. De igual manera desea agradecer al resto de instituciones, organizaciones y empresas que de una u otra forma colaboran en el éxito del Simposio.

La cuarta edición de SISOFT presenta las innovaciones tecnológicas y el estado actual de la investigación en las áreas de *Ingeniería de Informática* e *Ingeniería de Sistemas*, y en particular en Ingeniería de Software y en Sistemas de Información. Las comunicaciones aceptadas, junto con las conferencias magistrales, plenarias, talleres, seminarios, mesas redondas (paneles), mostrarán durante los tres días del Simposio, los temas más relevantes y candentes presentados por investigadores, docentes, profesionales, estudiantes de doctorado y maestría, etc., y una muestra del estado del arte en los temas centrales del mismo.

El contenido

Los dos volúmenes publicados recogen las casi ochenta comunicaciones aceptadas –de las más de cien presentadas- y que serán defendidas durante las sesiones del Simposio. Los artículos que se presentan en ambos volúmenes y las conferencias plenarias y magistrales que se impartirán tratarán sobre temas tan variados e innovadores como los siguientes:

- Nuevas metodologías de Ingeniería de Software y Sistemas de Información
- Arquitecturas *hardware*
- Supercomputadores
- Sistemas de Información Geográfica (GIS)
- Metodologías de Gestión de Proyectos como ITIL, PMP, .
- Gestión de calidad y métricas de software
- Gestión de la Información y el Conocimiento en Ciencias de la Salud, Bioinformática y Biotecnología
- Interacción Persona-Máquina y Usabilidad
- Ingeniería Web (Plataformas, Aplicaciones y Servicios Web)
- Estado actual de la industria del Software (software libre, software bajo demanda, videojuegos, ...)
- Estado del arte en Ingeniería de Software
- Web 2.0 (tecnologías, aplicaciones, Ajax, RSS,...)
- Redes Sociales Virtuales
- Agentes y Multiagentes
- Responsabilidad Social Corporativa: en sus enfoques empresarial y tecnológico
- Ingeniería multimedia y Realidad Virtual
- Gestión del Conocimiento e Inteligencia de Negocios
- Estado actual de la Sociedad de la Información y el Conocimiento
- Educación virtual, a distancia: *e-learning*
- Innovaciones tecnológicas en Tecnologías de la Información y las Comunicaciones (TIC)
- Almacenamiento virtual y en red
- Software avanzado de Microsoft (C#, ...)
- Innovaciones y mejoras de lenguajes de programación (Java, C++, C#, JavaScript, XML, Ajax,...)
- ...

Solo resta comentar que las conferencias, seminarios, talleres, mesas redondas, y conclusiones más sobresalientes se publicarán en un tercer volumen que verán la luz a la terminación del Simposio.

Los diferentes comités de SISOFT 2006 agradecen de nuevo a todas las instituciones, empresas, conferenciantes, ponentes, congresistas, ... su participación y quedan a su disposición durante todo el evento y los convocan para SISOFT 2007, pendiente de confirmación del país organizador, sede y fecha, pero que con toda seguridad seguirá la tradición científica e investigadora de las cuatro primeras ediciones

En Cartagena de Indias, agosto de 2006

Fdo: Dr. Luis Joyanes Aguilar

Presidente de SISOFT

Calidad de Productos de Software: Un estado del arte de la medición

Oswaldo Gómez, Hanna Oktaba
Universidad Nacional Autónoma de México,
Posgrado en Ciencias e Ingeniería de la
Computación,
México D.F., México, 04510
{oswaldog,ho}@uxmcc2.iimas.unam.mx

Mario Piattini, Félix García
Universidad de Castilla-La Mancha,
Departamento de Ciencias de la Computación,
Ciudad Real, España, 13071
{Mario.Piattini, Felix.Garcia}@uclm.es

ABSTRACT

It is a fact that software quality is one of the main company's objectives. Since quality is a critical factor of success in a very competitive market. Hence, software quality really constitutes a culture that affects all organizations. Quality is a systematic activity focuses on properties and characteristics of products and services. Therefore, it is necessary to establish methods, which assist us validating if such properties and characteristics are fulfilling our quality objectives. Software measures help us to reach that objective. Since software measures are useful for understanding, evaluating, improving and controlling software process and products. In this article we shall focus on products' software measurements. Based on a systematic review, we will show the attributes and entities that are measured nowadays, the measurements bias and how software measures support us in our quality assurance.

Keywords: Measures, quality, software products

RESUMEN

La calidad de Software es hoy en día uno de los principales objetivos de las empresas, porque es un factor crítico de éxito en un mercado muy competitivo; por lo tanto, la calidad realmente constituye una cultura que afecta a todas las organizaciones. La calidad es una actividad sistemática que se enfoca en propiedades y características de un producto o servicio. Así que es necesario establecer métodos que nos asistan en validar si tales características y propiedades cumplen con nuestros objetivos de calidad; las mediciones nos ayudan a alcanzar tal fin, ya que nos sirven para entender, evaluar, mejorar y controlar los productos y procesos de software. En este artículo nos enfocaremos en las mediciones para productos. Partiendo de una revisión sistemática, mostraremos los principales atributos y entidades que se miden actualmente, las tendencias de medición y como las medidas del software apoyan el aseguramiento de calidad.

Palabras claves: Medición, Calidad, Producto de Software

1. INTRODUCCIÓN

El software ha tenido numerosos problemas relacionados con cancelaciones de proyectos, retardos en las entregas, altos costes, insatisfacción del cliente, etc. Para cambiar esta situación, las organizaciones han tendido a mejorar la calidad de sus productos y servicios; por estas razones, la calidad de software ha ido evolucionando de sólo concentrarse en detectar errores a constituir toda una disciplina, una estrategia, una política y una cultura que rige la supervivencia de las empresas en un mercado extremadamente competitivo, como lo es la industria del software. Por lo tanto, el desarrollo de

sistemas informáticos debe ser más eficiente y la calidad de los productos debe ser alta, para satisfacer los requisitos del cliente de tal manera que se asegure el éxito.

El software tanto en su desarrollo como en su aplicación, conlleva una serie de requisitos de calidad que aseguran un funcionamiento consistente, para realizar lo que el cliente solicitó, pudiéndolo usar de manera natural, por estos motivos, las mediciones juegan un papel crítico en la calidad de los procesos y productos de sistemas informáticos, porque proveen de bases científicas a la ingeniería del software, convirtiéndola, poco a poco, en una verdadera disciplina de la ingeniería [24]. Actualmente las medidas han probado ser muy útiles para entender, evaluar y controlar tanto los atributos como las propiedades de los productos y procesos de software, permitiéndonos detectar oportunidades de mejora, entender que ocurre durante el desarrollo y el mantenimiento de los proyectos [20], optimizar los procesos y productos del software [9], desde una perspectiva de evaluación, seguimiento y predicción para soportar mejores decisiones [8], mostrando áreas problemáticas en la calidad de sistemas y más aún, nos ayudan a institucionalizar la mejora de proceso de software.

Este artículo se centra principalmente en medidas de software para productos, así que se realizó una revisión sistemática de literatura, con el fin de encontrar medidas, saber las tendencias de las mismas y, reconocer que atributos se han estado midiendo de los productos de software. Después de esta introducción, procederemos, en la siguiente sección, a explicar de manera muy breve los pasos seguidos en la revisión sistemática. Posteriormente, mostraremos los resultados obtenidos de dicha revisión para finalizar con las conclusiones y trabajos futuros.

2. REVISIÓN SISTEMÁTICA DE MEDICIONES

Primero que nada, debemos mencionar que nuestra *revisión sistemática* fue ejecutada siguiendo la propuesta de [15] y se propuso dar respuesta a la siguiente pregunta fundamental: ¿Cuáles son las mediciones más actuales y usadas en la literatura? Nuestro *protocolo de búsqueda* se desarrolló entorno a esta pregunta.

Una vez establecida la pregunta principal, realizamos las búsquedas con todas las combinaciones posibles de las siguientes cadenas: “(measure OR metric OR quality OR quantitative) AND (process OR engineering OR maintenance OR management OR improvement OR Software testing OR development)”, en los motores de búsqueda de las siguientes fuentes: ACM Digital Library, Search IEEE magazines, Wiley Interscience, and Science@Direct. Los resultados de la búsqueda en Internet son mostrados en la Tabla 1.

Fuentes	Resultados de Búsquedas	Revisados	Aceptados
Science@Direct	3569	78	10
ACM	950	85	28
IEEE	3740	111	32
Wiley	653	20	8
TOTAL	8912	294	78

Tabla 1. Resultados totales de Búsqueda

Como se puede apreciar en la Tabla 1, se obtuvo una cantidad considerable de resultados, pero extremadamente muy pocos de ellos se aceptaron, aproximadamente menos del 1%, esto se debe a que nuestros filtros de búsqueda no fueron muy restrictivos, lo cual nos dio la ventaja de no perder posibles estudios importantes. Por otra parte, nuestro *criterio de calidad* confió en las fuentes escogidas porque son serias, comprometidas con la calidad de sus artículos y fueron avaladas por expertos. De acuerdo con el *criterio de Inclusión y Exclusión* definido en nuestro *Protocolo de Revisión*, se aceptaron todos los artículos relacionados con mediciones para el desarrollo, mantenimiento, administración y gestión proyectos de software que contuvieran las cadenas de búsqueda en el título, las palabras claves y, en algunos casos, en todo el documento.

Posteriormente se realizaron *extracciones objetivas y subjetivas* de la información contenida en los estudios, dichas extracciones de datos fueron realizadas elaborando una plantilla de extracción, con el fin de documentar los resultados de este proceso, siguiendo la propuesta de [25] que se muestran en la Tabla 2.

Resultados de la Extracción Objetiva	
Identificación del estudio	
Nombre	
Autor	
Institución	
Revista	
Fecha	
Metodología del Estudio	
Resultados del Estudio	
Problema del Estudio	
Resultados de la extracción subjetiva	
Información por los autores	
Resúmenes e impresiones generales	

Tabla 2: Plantilla para la extracción subjetiva y objetiva de los datos.

Por la gran cantidad de tópicos encontrados y por nuestro propósito de resumir la información de la mejor manera posible, nos basamos en los conceptos definidos en la *Ontología de Medición de Software* propuesta en [10], que tiene como objetivo contribuir a la armonización de las diferentes medidas de software y estándares, proveyendo un grupo coherente de conceptos comunes usados en la medición del software. Así que, apoyados por dicha ontología, elaboramos una plantilla para clasificar la información extraída de las siguientes tres formas: “Qué se mide”, “Cómo se mide” y “Cuándo se mide”.

Para definir “Qué se mide”, la ISO 15504 [13], CMM [22] y CMMI [5] proponen marcos de calidad para mejorar los niveles de madurez del proceso de desarrollo de software, definiendo como entidades a ser medidas, para alcanzar tal fin, al *Proyecto*, al *Proceso* y al *Producto*. Por lo que extrajimos las características medidas de dichas entidades y las clasificamos en atributos y subatributos [9] internos y externos (Tabla 2). Cabe destacar aquí, que este trabajo en particular se enfocará únicamente en las mediciones de los Productos y su impacto en la calidad del software.

Qué se mide						
Entidades			Atributos	Subatributos	Tipo de Atributo	
Proyecto	Proceso	Producto			Interno	Externo

Tabla 2. Definición de entidades

Una vez que descubrimos que se mide, procedimos a analizar el punto de “Cómo se mide”, clasificando las mediciones extraídas de los artículos en términos de las siguientes características: Representación, descripción, medida base, derivada o indicadores [9,10,12], escala [9,24] y validación empírica [1,14,19,29] o teórica [3,21,27,28,30]. La Tabla 3 resume esto.

Cómo se mide					
Representación	Descripción	Medición		Escala	Validación
		Básica	Derivada		

Table 3: Definition of measure attributes

Finalmente analizamos “Cuándo se mide”, utilizando el ciclo de vida de software en cascada, para mostrar la cantidad de medidas contenidas en las fases del mismo. Escogimos el ciclo de vida en cascada dada su genericidad, a la hora de clasificar las medidas que se obtienen en cada fase, de forma que esta clasificación no fuera dependiente de una metodología concreta de desarrollo y/o mantenimiento.

3. ANÁLISIS DE RESULTADOS

Con esta revisión sistemática, pudimos observar que la entidad más medida es el producto software, ya que las medidas del producto se centran en su calidad y en aspectos técnicos, de los cuales se tienen atributos mejor definidos que en el proceso y en el proyecto, además las medidas de producto constituyen la base para las medidas de estas dos entidades. Por otra parte, debemos destacar que las mediciones para producto son mayores debido a que la calidad tuvo un fuerte enfoque en el producto desde sus comienzos y, posteriormente, se ocupó del proceso.

También analizamos la cantidad de medidas hechas a los atributos encontrados en los artículos seleccionados. El tamaño es uno de los atributos más medidos, no sólo porque es una medición básica usada en la mayoría de las mediciones derivadas enfocadas, principalmente, al control de proyectos y la estimación de defectos, si no también porque una vez medido el tamaño, podemos medir el tiempo del desarrollo de sistemas y, una vez calculado el tiempo, podemos hacer la estimación del costo, por lo que el tamaño constituye la base de la mayoría de las mediciones de control.

Por otra parte, hemos encontrado que la complejidad [18,11], se ha usado en varios contextos, por ejemplo: en el diseño, arquitectura, código fuente, diagramas de UML, etc., por lo tanto, podemos ver en la Figura 2, que este atributo fue otro de los que acumuló una gran cantidad de mediciones de sus diferentes aplicaciones. También, pudimos observar que existen mediciones de atributos tanto para la programación Orientada a Objetos (OO) [4,16,17] como para la programación WEB[6], por lo que existe una gran cantidad de medidas para estos dos enfoques de programación.

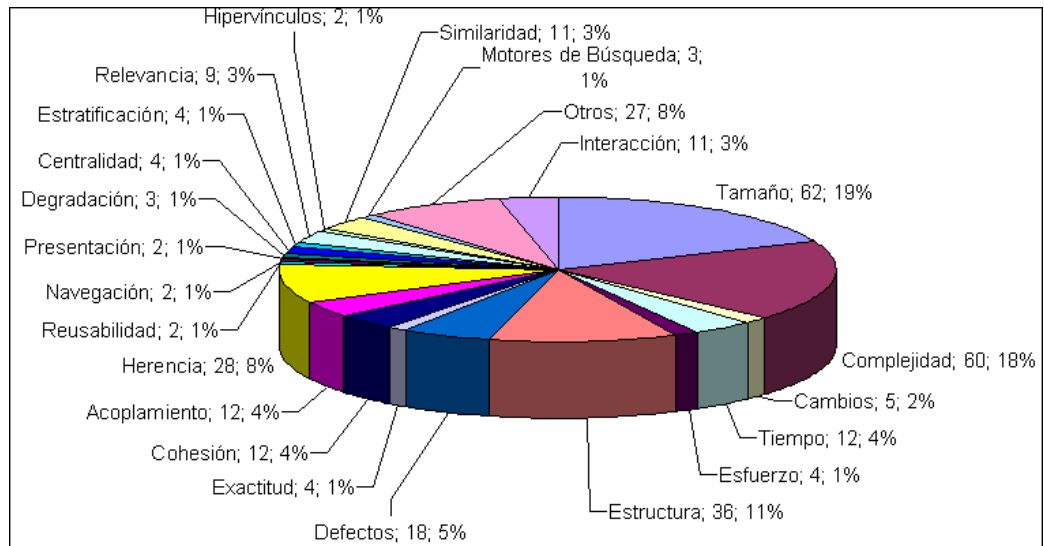


Figura 2. Atributos medidos

Ahora, debemos señalar que las medidas relacionadas con la calidad del producto, abarcaron la estimación de defectos, errores encontrados en alguna fase del ciclo de vida del software, efectividad en la detección de errores, densidad de fallos, estimación de la confiabilidad y defectos en ejecución durante las pruebas; evaluando, principalmente, los siguientes atributos de calidad: seguridad, portabilidad, manejabilidad, robustez, usabilidad y desempeño [24]. De lo anteriormente dicho, podemos deducir que contar defectos es la técnica más aplicada y aceptada para determinar la calidad del software.

Continuando con el análisis de los atributos más medidos, en este punto, nos enfocaremos en unos de los aspectos más importantes de la medición concerniente a la validación. El objetivo de las validaciones teóricas es verificar si la idea intuitiva de los atributos a medir es considerada en la definición de la medición y, por otra parte, el objetivo de la validación empírica, es obtener información objetiva concerniente a la utilidad de las mediciones propuestas. La validación teórica por sí misma no es suficiente para garantizar la utilidad de una medición, ya que puede ocurrir que una medida tenga una validación teórica, pero que no tenga una relevancia práctica en relación con un problema especificado. Por lo tanto, en la Figura 2, podemos observar que la mayoría de las mediciones tienen validaciones empíricas o ambas.

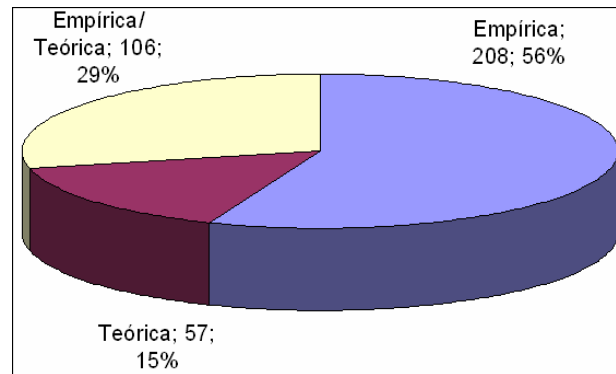


Figura 2. Proporción de validaciones a las mediciones.

Otro punto a tratar en este trabajo, se refiere a las mediciones en función al enfoque de programación y de medición en que se han usado. Las medidas más representativas de los modelos conceptuales [23] encontradas fueron para los modelos de Bases de Datos Relacionales, para diagramas de Clases, de Casos de Uso y de Estados del Lenguaje de Modelado Unificado (UML) [17]. También descubrimos que la gran popularidad de los proyectos OO [4,16,17], se refleja en la cantidad de medidas hechas en ese campo, ya que representa casi la mitad de las existentes en la literatura

actual, además las medidas de tamaño funcional como los Puntos de Función (FP) propuestos por la IFPUG [12], los Objetos Función para el paradigma Orientado a Objetos y usados en el modelo de estimación COCOMO II [2] o los Puntos de Función Totales (FPF) elaborados por COSMIC[7], han tenido gran aceptación para las estimaciones de tamaño de sistemas y consecuentemente han constituido la base para la estimación de costos y calendarios para varias organizaciones.

Un aspecto importante encontrado en esta revisión, es el esfuerzo por lograr una definición más universal de las mediciones para sistemas Web [6], estableciendo marcos de referencia y modelos para clasificar las medias en este campo, ya que las aplicaciones Web, son generalmente consideradas entre las más difíciles de construir, tanto en costo como en tamaño, por estos motivos, las mediciones para ese campo abarcan casi la cuarta parte del total.

Para concluir con esta parte del análisis, nosotros agrupamos dentro del paradigma imperativo, aquellas medidas de atributos referentes a la estructura de un programa, que pudieran ser utilizadas en cualquier paradigma de programación y que no se especializan en alguno en particular, ya que el paradigma imperativo hace uso de la secuencia, selección e iteración para escribir programas, siendo estos aspectos muy generales y, por lo tanto muy usados, como consecuencia este tipo de medidas ocupan un 22 % del total de las encontradas en esta revisión. La Figura 3 resume esta parte del análisis.

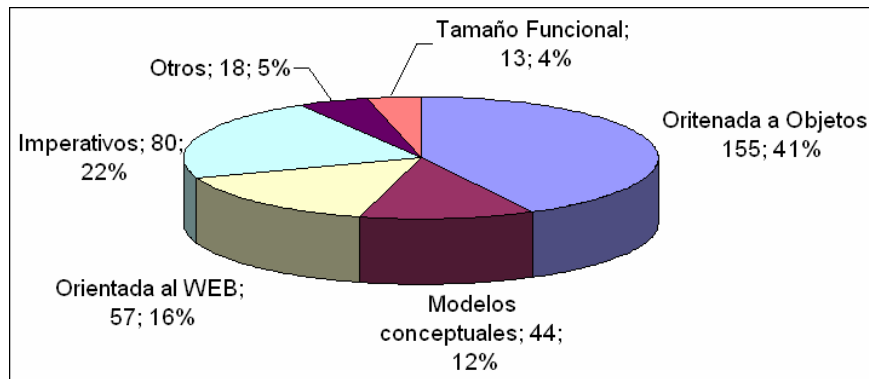


Figura 3. Mediciones y enfoques de programación.

Finalmente, analizaremos “Cuándo se mide”. Debemos resaltar, de la Figura 4, que en la fase de diseño, productos tales como la arquitectura, el diseño del sistema, el análisis de requisitos etc., son generados y constituyen lo que será el sistema de software, por lo tanto, es necesario que, en esta fase, se realicen mediciones que nos ayuden a conocer características de dichos productos. Por otra parte, se debe destacar que productos tales como manuales, código fuente, el sistema en sí y la mayor parte de los productos del software son generados en la fase de desarrollo, siendo posible recolectar información cuantitativa de los mismos, por lo que esta fase es también una de las más medidas. Además, de acuerdo a PSP [26], mediciones como el tamaño, tiempo y defectos son comúnmente recolectados en esta fase.

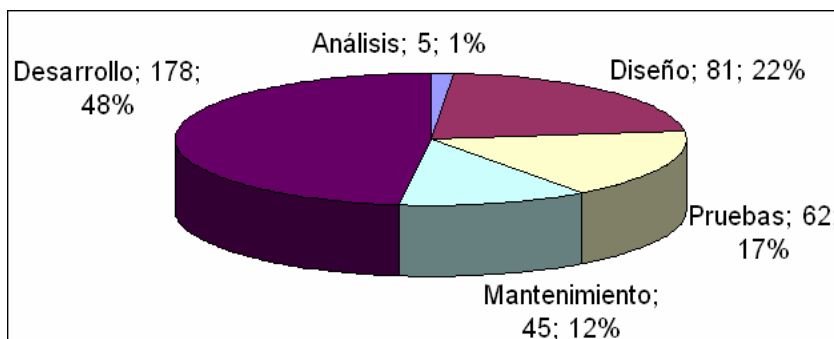


Figura 4. Mediciones de las fases del ciclo de vida

La calidad es uno de los puntos más importantes para el control dentro de la administración de un proyecto, porque nos ayudan a validar si nuestros productos de software cumplen con los requisitos establecidos, dichos requisitos deben ser

especificados numéricamente [8], La calidad no sólo se considera en la fase de pruebas, si no en todo el proceso de desarrollo, ya que los orígenes de las fallas se pueden encontrar, incluso en la fase de diseño. Además, basados en reportes de fallos durante todo el proceso, las mediciones determina la validez de las predicciones de calidad; como resultado de esto, se pueden determinar defectos en módulos y tasas de error durante la fase de pruebas. Resulta más rápido y barato concentrar el esfuerzo de buscar defectos durante las primeras fases del desarrollo, detectar y corregirlos lo más pronto posible en el ciclo de vida del software, particularmente en esta revisión hemos encontrado que medidas de calidad son recolectadas principalmente en la fase de diseño y de Desarrollo.

4. CONCLUSIONES

La percepción de la calidad es uno de los factores con mayor influencia para las compañías, ya que siempre se enfrentan con el dilema de sostener una relación con el cliente en la que, por entregar a tiempo un producto de software con mala calidad, arruinan su reputación o, inútilmente, gastan una gran cantidad de tiempo extra -con los costos asociados- para mejorar la calidad del producto de software a un nivel que nadie quiere pagar. El óptimo estado es uno intermedio, esto significa alcanzar un nivel correcto de calidad y tiempo de entrega, lo que implica conocer continuamente qué representa realmente este nivel de calidad.

Las mediciones del software son totalmente imprescindibles para establecer el nivel correcto de calidad antes mencionado, ya que nos apoyan en el control de proyecto, la evaluación, la mejora y el mantenimiento de los productos y procesos del software, desde una perspectiva de seguimiento y predicción. Las medidas nos ayudan a tomar mejores decisiones, también cabe destacar aquí que existe una fuerte tendencia, en la literatura actual, por obtener información objetiva concerniente a la utilidad de dichas mediciones, es decir, a validarlas empíricamente y, de esta manera, dar una garantía para alcanzar el equilibrio antes mencionado.

En este estudio nos percatamos que el tamaño, es una de las medias básicas más importantes dentro del desarrollo de productos de software, porque una vez medido el tamaño, podemos medir el tiempo del desarrollo de sistemas y, una vez calculado el tiempo, podemos hacer la estimación del costo. Por otra parte, aproximaciones metodológicas, han enfatizado que el primer paso es reconocer que todos los requisitos de calidad pueden y deben ser especificados numéricamente, esto significa que se deben cuantificar atributos tales como la seguridad, la portabilidad, la adaptabilidad, la mantenibilidad, la robustez, la usabilidad, la confiabilidad y el desempeño.

La calidad no sólo se considera en la fase de pruebas, si no en todo el proceso de desarrollo. Buscando defectos desde las fases tempranas de ciclo de vida del proyecto. En la fase de diseño se generan productos que constituyen lo que será el sistema de software, por otra parte, la mayor cantidad de los artefactos son generados en la fase de desarrollo, por lo tanto, en estas dos fases resultaron tener también una buena cantidad mediciones, que ayudan a conocer características de sus productos, siendo el conteo de defectos y la confiabilidad las mediciones más aplicadas y aceptadas para determinar la calidad de los mismos.

Finalmente debemos resaltar que existe una fuerte tendencia a medir proyectos OO y proyectos Web, ya que esta clase de proyectos son hoy en día los más populares en el mercado, por lo que se han realizado esfuerzos para lograr una definición más universal de sus mediciones y, por consiguiente, de sus características de calidad, estableciendo marcos de referencia y modelos para clasificar las medias en este campo.

REFERENCIAS

- [1] Basili V.y Weiss D., “A Methodology for Collecting Valid Software Engineering Data.”, *IEEE Transactions on Software Engineering*. (1984), 10, pp. 728-738.
- [2] Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D. and Steece, B. (2000). *Software Cost Estimation with COCOMO II*. Prentice-Hall, (Julio, 2000).
- [3] Briand L., Morasca S. and Basili V., “Property-Based Software Engineering Measurement”, *IEEE Transactions on Software Engineering*. 22(1), pp. 68-86, (1996).
- [4] Chidamber S. and Kemerer C., “A Metrics Suite for Object Oriented Design”, *IEEE Transactions on Software Engineering*. 20(6), pp. 476-493, (1994).
- [5] Chrissis M. B., Konrad M. Shrum Sandy, *CMMI®: Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 1st Edition, Carnegie Mellon Software Engineering Institute, USA, (2003).

- [6] Calero, C., Ruiz, J., Piattini, M., 2005, Classifying web metrics using the web quality model. *Online Information Review*. Vol 29 No 3, pp. 227-248
- [7] COSMIC. *COSMIC Measurement Manual*. The COSMIC Implementation Guide for ISO/IEC 19761:2003, (2003).
- [8] Ebert C., Dumke R., Bundschuh M., Schmietendorf A., *Best Practices in Software Measurement How to use metrics to improve project and process performance*, 295 Seiten-Springer 1st Edition, Berlin, (2004).
- [9] Fenton N. and Pfleeger S. L., *Software Metrics: A Rigorous & Practical Approach*, PWS Publishing Company, Second Edition, (1997).
- [10] García F., Bertoa M. F., Calero C., Vallecillo A., Ruíz F., Piattini M., Genero M., “Towards a consistent terminology for software measurement”, *Information and Software Technology*. xx, pp. 1-14, (2005).
- [11] Henry S. and S. Kafura, “Software Structure Metrics Based on Information Flow”, *IEEE Transactions on Software Engineering*. 7(5), pp. 510-518, (1981).
- [12] IFPUG, *IFPUG: Function Point Counting Practices Manual, Release 4.2*. International Function Point Users Group, USA –IFPUG, Mequon, Wisconsin, USA, (2004).
- [13] ISO/IEC 15504-2:2003, *Software engineering — Process assessment — Performing an assessment*, International Organization for Standardization, Geneva, (2001).
- [14] Juristo N. and Moreno A., *Basics of Software Engineering Experimentation*, Kluwer Academic Publishers, (2001).
- [15] Kitchenham B., *Procedures for Performing Systematic Reviews*, Joint Technical Report, Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia, pp. 1-28, (2004).
- [16] Lorenz M., y Kidd J., *Object-Oriented Software Metrics: A Practical Guide*, Prentice Hall, Nueva Jersey, USA, (1994).
- [17] Marchesi M., “OOA Metrics for the Unified Modeling Language”, *2nd Euromicro Conference on Software Maintenance and Reengineering*, pp. 67-73, (1998).
- [18] McCabe T., “A Software Complexity Measure”, *IEEE Transactions on Software Engineering*, 2, pp. 308-320, (1976).
- [19] Perry D., Porte A. and Votta L., *Empirical Studies of Software Engineering: A Roadmap. Future of Software Engineering*, Ed. Anthony Finkelstein, ACM, pp. 345-355, (2000).
- [20] Piattini M. y García F., *Calidad en el desarrollo y mantenimiento de software*, Ra-Ma, Spain, 1st Edition, (2003).
- [21] Poels G. y Dedene G., Distance-based software measurement: necessary and sufficient properties for software measures. *Information and Software Technology*, 42(1), pp. 35-46. (2000).
- [22] Raynus J., *Software Process Improvement with CMM*, Artech House, USA, 1st Edition, (1999).
- [23] Reynoso L., Genero M. y Piattini M., Measuring OCL Expressions: An Approach Based on Cognitive Techniques, Chapter 5 in “Metrics for Software Conceptual Models”, (Eds. Genero M., Piattini M. and Calero C.). Imperial College Press, UK. (2004).
- [24] Stephen H. K., *Metric and Models in Software Quality Engineering*, Addison-Wesley 2nd Edition, Boston, USA (2002)
- [25] Travassos G. H., Boilchi, J., Mian, P. G., Natali, A. C. C., *Systematic Review in Software Engineering*. Technical Report Programa de Engenharia de Sistemas e Computação PESC, Systems Engineering and Computer Science Department COPPE/UFRJ, Rio de Janeiro, pp. 1-30. (2005).
- [26] Watts S. Humphrey, *PSP A Self-Improvement Process for Software Engineers*, Addison-Wesley, (2005).
- [27] Weyuker E., “Evaluating Software Complexity Measures”, *IEEE Transactions on Software Engineering*. 14(9), pp. 1357-1365, (1988).
- [28] Whitmire S., *Object Oriented Design Measurement*, John Wiley & Sons, Inc, (1997).
- [29] Wohlin, C., Runeson, P., Höst, M., Ohlson, M., Regnell, B. y Wesslén, A., 2000. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers.
- [30] Zuse H., *A Framework of Software Measurement*, Berlin. Walter de Gruy, (1998).