

# MODELOS Y MÉTRICAS PARA EVALUAR CALIDAD DE SOFTWARE

Estayno, Marcelo<sup>(1)</sup>; Dapozo, Gladys<sup>(2)</sup>; Cuenca Pletch Liliana<sup>(3)</sup>, Greiner, Cristina<sup>(2)</sup>

(1) Departamento de Informática. Facultad de Ingeniería. Universidad Nacional de Lomas de Zamora  
mestayno@fibertel.com.ar

(2) Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura  
Universidad Nacional del Nordeste  
{gndapozo, cgreiner}@exa.unne.edu.ar

(3) Departamento de Ingeniería en Sistemas de Información. Facultad Regional Resistencia  
Universidad Tecnológica Nacional  
cplr@frre.utm.edu.ar

## CONTEXTO

Las líneas de I/D presentadas en este trabajo forman parte de las actividades definidas en el marco del proyecto “Modelos y métricas para la evaluación de la calidad de software orientados a Pymes”, presentado para su acreditación ante la Secretaría General de Ciencia y Técnica de la Universidad Nacional del Nordeste (UNNE). En este proyecto participan tres unidades académicas del país, los integrantes pertenecen a la Universidad Tecnológica Nacional Regional Resistencia (UTN-FRRe) y a la Facultad de Ciencias Exactas de la UNNE, bajo la dirección de un docente investigador de la Universidad Nacional de Lomas de Zamora (UNLZ).

El objetivo fundamental del proyecto es contribuir a la mejora en la calidad de los productos de software mediante modelos y métricas aplicados al producto y al proceso de creación, diseño, desarrollo y mantenimiento de software, como medio para aumentar la competitividad de las PYMES de la región NEA en el contexto de la industria del software.

## RESUMEN

En este trabajo se describen las líneas de investigación y desarrollo que se realizarán en el marco de un proyecto mayor referido a calidad de software orientado a Pymes de la región NEA del país, en el contexto de las iniciativas gubernamentales de promoción de la industria del software. La calidad del software, de por sí compleja y subjetiva, se basa en modelos y métricas que intentan realizar la medición de los distintos aspectos que afectan el proceso de desarrollo y el producto software. Los resultados de la medición y evaluación de la calidad aportarán al mejoramiento del software que se elabora y, consecuentemente, al incremento de la competitividad de quienes desarrollan, como así también, y principalmente, mejorar las aplicaciones y sistemas, y sus ámbitos de aplicación. En particular, en este proyecto se plantea además, estimular la vinculación de las universidades y las empresas Pymes de la región NEA, como así también, el mejoramiento de sus recursos humanos, como forma de contribuir al desarrollo local.

**Palabras clave:** Ingeniería de software, calidad del software, modelos y métricas de evaluación, calidad de aplicaciones web, programación orientada a objetos, programación orientada a aspectos, técnicas metaheurísticas.

## 1. INTRODUCCION

En la actualidad, gracias a los avances de la Informática, el software se encuentra en casi todos los campos de la actividad humana: la industria, el comercio, las finanzas, el gobierno, la salud, la educación, las artes, etc. La dependencia de las organizaciones respecto de este recurso es crítica. La automatización de las actividades y la generación y disponibilidad de información para la toma de decisiones, logradas a través del software, son claves para el logro de los objetivos y supervivencia de las organizaciones.

Por las razones expuestas, existe una creciente preocupación por lograr que los productos software cumplan con ciertos criterios de calidad. Para ello, se avanza en la definición e implementación de estándares que fijan los atributos deseables del software de calidad, a la vez que surgen modelos y metodologías para la evaluación de la calidad.

Por otra parte, en los últimos años el sector TIC ha tomado una relevancia particular en la Argentina, y se está asentando como un nuevo pilar del desarrollo nacional. La concentración geográfica de empresas, universidades y todos los actores involucrados en dicho sector, incrementa la utilización de la capacidad instalada de las regiones, potencia su industria y mejora la calidad de sus productos e insumos [1]. En consonancia con estos propósitos, a partir del año 2005, se constituyen en la región, primero el Polo IT Chaco y luego el Polo IT Corrientes, buscando mediante la asociatividad el crecimiento individual y conjunto de las empresas que lo componen.

### Calidad del software

La calidad del software es una compleja combinación de factores, que variarán entre diferentes aplicaciones. Diversos autores como Pressman [2], McCall [3] y estándares, como ISO 9126 han tratado de determinar y categorizar los factores que afectan a la calidad del software.

Una definición amplia de calidad, planteada en la norma UNE-EN ISO 8402 [4], expresa que “la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas”. Llevada esta definición al campo de la ingeniería de software, la IEEE Std 610 [5], señala que “la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Por su parte Pressman [2], se refiere a la calidad del software como “la concordancia con los requisitos funcionales y de rendimientos explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se espera de todo software desarrollado profesionalmente”.

Sommerville [6] sostiene que la calidad del software es un concepto complejo que no es directamente comparable con la calidad de la manufactura de productos. En la manufacturación, la noción de calidad viene dada por la similitud entre el producto desarrollado y su especificación. En un mundo ideal, esta definición debería aplicarse a todos sus productos, pero, para sistemas de software, existen cuestiones específicas que impiden aplicar este mecanismo.

### **Calidad del Producto y del Proceso**

Al intentar definir el concepto de calidad del software se debe diferenciar entre la calidad del Producto de software y la calidad del Proceso de desarrollo del mismo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto [7].

Hay un vínculo claro entre la calidad del proceso y del producto en producción debido a que el proceso es relativamente fácil de estandarizar y monitorizar. Cada sistema de producción se calibra, y debe producir una y otra vez productos de alta calidad. Sin embargo, el software no se manufactura, sino que se diseña. El desarrollo de software es un proceso más creativo que mecánico, donde las experiencias y habilidades individuales son importantes. La calidad del producto, sea cual fuere el proceso utilizado, también se ve afectada por factores externos, como la novedad de una aplicación o la presión comercial para sacar un producto rápidamente.

### **Modelos de calidad de software**

A lo largo del tiempo se han desarrollado diferentes modelos para evaluar la calidad del software, que intentan descomponer la calidad en una categoría de características más sencillas. Entre ellos puede

mencionarse el de McCall, Evans y Marciniak, Deutch y Willis, FURPS, entre otros [8].

Un hito en la definición de estándares de calidad de producto software, lo constituye la publicación del ISO9126 en el año 1991. Luego, en el año 2001, este estándar fue reemplazado por dos estándares relacionados: el ISO/IEC 9126, que especifica características y métricas de la calidad del software; y el estándar ISO/IEC 14598, que especifica la evaluación de productos de software [9].

El estándar ISO/IEC 9126 se compone de cuatro partes: modelo de calidad [10], métricas externas [11], métricas internas [12] y métricas para la calidad en uso [13]. Propone un modelo de calidad categorizando la calidad de los atributos software en seis características (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), las cuales son subdivididas en subcaracterísticas. La calidad de uso es definida como “la capacidad del software que posibilita la obtención de objetivos específicos con efectividad, productividad, satisfacción y seguridad” [14].

El modelo más actual está representado por las normas ISO 25000:2005, conocidas con el nombre de SQuARE (*Software Quality Requirements and Evaluation*), basada en ISO 9126 y en ISO 14598, se desagrega en 5 tópicos: 1-Gestión de la Calidad (2500n), 2- Modelo de Calidad (2501n), 3- Medidas de Calidad (2502n), 4- Requerimientos de Calidad (2503n) y 5-Evaluación de la Calidad (2504n) [8].

La especificación de requisitos de calidad y la evaluación de productos software son dos procesos que por su inherente complejidad pueden beneficiarse del proceso que regule su realización. Sin embargo, y como señala el estándar SQuARE, es importante que sus objetivos estén alineados. Por ello, la creación de una norma que regule su realización pueda ser muy beneficiosa, en cuanto a la consistencia de los resultados obtenidos [14].

Otro aspecto destacable de SQuARE es la incorporación de una normalización de la terminología, considera la Metrología como la ciencia de la medida y la necesidad de amoldar los conceptos usados en Ingeniería del Software a los utilizados en otras disciplinas que hacen uso de la medición [15].

### **Métricas y Medición.**

La medición es un elemento clave en cualquier proceso de ingeniería. Las medidas se emplean para comprender mejor los atributos de los modelos que se crean y evaluar la calidad de los productos de la ingeniería. Por las características inherentes al software, sus medidas y métricas son indirectas y, por lo tanto, expuestas al debate [2].

Una métrica contiene la definición de un método de medición o un método de cálculo y la escala asociada.

El método de medición es la secuencia lógica particular de operaciones y posibles heurísticas, especificada para permitir la realización de la

descripción de una métrica por una actividad de medición. Por otro lado, la escala se define como un conjunto de valores con propiedades definidas.

La propiedad más importante de una escala es su tipo, considerando que puede ser Categórica o Numérica. A su vez, dependiendo de la naturaleza de la relación entre los componentes de la escala, pueden clasificarse en: nominal, ordinal, intervalo, proporción o absoluta. El tipo de escala de los valores medidos define las transformaciones admisibles y afecta las operaciones matemáticas y estadísticas que pueden ser aplicadas.

Las métricas pueden ser directas, sobre las que puede aplicarse un método de medición (objetivo o subjetivo); o indirectas, que son aquellas definidas en función de otras métricas y se calculan en base al método de cálculo asociado, es decir en base a una fórmula.

### **Calidad en aplicaciones Web**

Los avances en Internet han conducido a un desarrollo impactante de sistemas y aplicaciones basadas en la Web, suceso que se presenta como el más significativo en la historia de la Computación

Muchas de las nuevas tecnologías y estándares de la Web han surgido en los últimos años para mejorar el apoyo a nuevas aplicaciones Web: XML, servicios Web, Web semántica, técnicas de personalización de la Web, minería Web, inteligencia, *contextaware* y móviles y servicios Web.

Las aplicaciones web son diferentes de otras categorías de software; son eminentemente de red, las gobiernan los datos y se encuentran en evolución continua. La inmediatez dirige su desarrollo, la seguridad es un requisito prioritario y la demanda de estética, así como la entrega de contenido funcional, son factores diferenciales adicionales [2].

El estudio de la calidad de productos y procesos de desarrollo para la Web es muy reciente y todavía no se dispone de métodos de evaluación ampliamente difundidos para este tipo de entorno, por lo tanto, existe la necesidad de metodologías efectivas para la obtención de aplicaciones Web de calidad.

La Ingeniería Web surge debido a la necesidad de lograr enfoques disciplinados y nuevos métodos y herramientas para desarrollar, desplegar y evaluar los sistemas y aplicaciones basados en la Web. Estos enfoques y técnicas deben considerar las particularidades del nuevo medio, el contexto y los escenarios operativos y, principalmente, la diversidad de perfiles de usuarios que constituyen desafíos adicionales al desarrollo de aplicaciones Web [2].

Cómo se mide la calidad del software para la web? En general, con los mismos modelos que para el software tradicional. Sin embargo, hay características que son más relevantes en este contexto, como por ejemplo, la facilidad de uso, funcionalidad, confiabilidad, eficiencia y facilidad de mantenimiento. Olsina [16] define un “árbol de requisitos de calidad” para aplicaciones Web y

Offut [17] agrega otros atributos como Seguridad, Disponibilidad, Escalabilidad, Tiempo en el Mercado.

### **Calidad en Programación Orientada a objeto**

El desarrollo de programas orientados a objetos (POO) es cada vez mayor, sin embargo, no ha evolucionado al mismo ritmo el uso de métricas para este paradigma. Las métricas orientadas a objetos, al igual que las del software convencional, buscan poder entender mejor la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo llevado a cabo al nivel del proyecto.

Sin embargo, la POO difiere en importante medida del desarrollado utilizando enfoques tradicionales. Por esta razón las métricas deben ajustarse a las características que lo distinguen, como ser encapsulamiento, ocultamiento de información, herencia y técnicas de abstracción de objetos que hagan única a esa clase.

Entre las métricas encontradas en la literatura que han tenido relevancia en la orientación a objetos sobresalen las definidas por Abreu, Chidamber y Kemerer y Lorenz y Kidd, que abordan todos los posibles niveles de granularidad y características en sistemas OO, como ser: Métricas a nivel de sistema, Métricas a nivel de acoplamiento, Métricas a nivel de herencia, Métricas a nivel de clases, Métricas a nivel de métodos [18].

### **Calidad en Programación Orientada a Aspectos**

En el modelo de Programación Orientado a Objetos (POO) toda tarea específica debe ser responsabilidad de una clase o de un pequeño número de clases agrupadas de alguna forma lógica. Sin embargo, existen ocasiones en las que determinados servicios se utilizan en diversas clases y no tienen suficiente entidad para incluirlos en una clase específica, lo que provoca repetición de código a lo largo de toda la aplicación. Por ejemplo, los bloques de código dedicados a la sincronización de los accesos a los recursos, a la persistencia de los datos, al registro de auditorías (logs), etc. Estos bloques de código son características o temas de interés (*concerns*) dentro del sistema software. La diseminación de estos concerns a través de varias clases son conocidos como *crosscutting concerns*.

Normalmente los aspectos están mezclados en los principales módulos de los componentes del sistema causando el problema de tener código desordenado. La Programación Orientada a Aspectos (POA) trata de encapsular estas características en módulos en vez de tenerlos dispersos en los componentes del sistema.

En sus inicios, la POA se centró principalmente en el nivel de implementación y codificación, pero en los últimos tiempos esta separación se está llevando a cabo a nivel de diseño. Esto impacta positivamente en la calidad integral del software.

Dado el continuo desarrollo de aplicaciones que contemplan este nuevo concepto, diferentes modelos de métricas han aparecido como una forma de evaluación de la calidad de las mismas [19] [20] [21]. Si bien estos nuevos modelos tienen características similares a los modelos de métricas para sistemas orientados a objetos, tienen también características particulares a los aspectos.

Las métricas sobre las cuales se trabaja consideran: Cantidad de Aspectos, Cantidad de relaciones existentes entre aspectos y una clase, Cantidad de clases relacionadas con un mismo aspecto, Cantidad de puntos de enlace en una clase, Cantidad de Clases Tejidas.

La POA es un nuevo paradigma que aún adolece de madurez y formalidad, por lo que las líneas de investigación apuntan a definir métricas que reúnan las mejores características de las existentes, entre otras.

### **Técnicas metaheurísticas orientadas a la calidad del proceso de desarrollo**

Con el aumento de las prestaciones de las computadoras y recursos asociados el desarrollo de Software se hizo más complejo, pasando de ser una tarea realizada por una sola persona en pocas horas a convertirse en un conjunto de actividades interrelacionadas que deben realizarse en grandes equipos de trabajo durante meses.

Los problemas de optimización se plantean todas las ingenierías y la Ingeniería del Software, a pesar de ser una disciplina joven, no es una excepción. Según Chicano [22] en la actualidad existe un creciente interés por aplicar técnicas de optimización a problemas de Ingeniería del Software, ya sea mediante el uso de algoritmos exactos, heurísticos ad hoc o metaheurísticos. Si bien los algoritmos exactos garantizan encontrar el óptimo global de cualquier problema, tienen el grave inconveniente de que en problemas reales su tiempo de ejecución crece de forma exponencial con el tamaño del problema. Los algoritmos heurísticos *ad hoc*, en cambio, suelen ser bastante rápidos pero las soluciones no suelen ser óptimas. Las metaheurísticas ofrecen un equilibrio adecuado entre ambos extremos: son métodos genéricos que ofrecen soluciones de buena calidad en un tiempo moderado. En esta línea de investigación se analizan las alternativas para aplicar, a los problemas de optimización en Ingeniería de Software, algoritmos metaheurísticos que ofrezcan soluciones de cierta calidad en un breve periodo de tiempo: un compromiso entre calidad de la solución y rapidez. Existen numerosos trabajos en este sentido, que abordan distintas etapas del desarrollo de Software: Des Creer y Ruhe [23] abordan el problema de la selección de requisitos para cada iteración del proceso de desarrollo, Simons y Parmee [24] plantean el diseño conceptual como un problema de optimización. El problema de la generación automática de código paralelo óptimo a partir de

código secuencial ha sido abordado por Nisbet [25] y Williams [26]. Soluciones a los problemas de optimización relacionados con la planificación de proyectos y la generación automática de casos de prueba han sido propuestas por Enrique Alba et al en [27] y [28]. En [22] Chicano presenta un análisis de la evolución de este campo de investigación, demostrando un aumento casi lineal de los trabajos a partir de 1999. El mismo informe demuestra que el mayor interés de la comunidad científica se ha centrado en la fase de pruebas, seguida por la fase de gestión de proyectos. Esto podría deberse, según el autor, a que aproximadamente la mitad del tiempo de un proyecto software y más de la mitad de su costo se dedica a la fase de pruebas. Se trata entonces de un campo de investigación que es necesario explorar para definir las áreas de vacancia donde sea posible realizar aportes sustantivos con relevancia regional.

## **2. LINEAS DE INVESTIGACION y DESARROLLO**

- Análisis, estudio y discusión de modelos de evaluación de calidad, estándares y metodologías.
- Relevamiento de metodologías y herramientas utilizadas por las empresas Pymes de la región para la evaluación de la calidad de sus productos software y la calidad del proceso de desarrollo de software.
- Análisis y estudio de modelos y métricas de evaluación de calidad de uso de aplicaciones Web. Propuesta metodológica para la aplicación de un modelo adaptado a las características de las Pymes locales.
- Relevamiento de modelos y métricas de evaluación de calidad de aplicaciones orientadas a objetos. Aplicación y comparación de, por lo menos, dos metodologías relevantes.
- Análisis y estudio de modelos y métricas de evaluación de calidad de aplicaciones orientadas a objetos. Propuesta de un modelo para evaluar aplicaciones orientadas a aspectos.
- Identificación de problemas de optimización en Ingeniería del Software.
- Descripción, formalización, aplicación y evaluación de técnicas metaheurísticas para resolver problemas de optimización de la Ingeniería del Software.
- Selección, descripción y formalización de los problemas que se detecten como relevantes para el campo científico y para el desarrollo regional.

## **3. RESULTADOS ESPERADOS**

Dentro de las líneas de trabajo mencionadas, se espera obtener los siguientes resultados:

- Conocimiento actualizado de los distintos modelos, métricas, estándares y herramientas de evaluación de calidad de software para su

aplicación al ámbito de las PYMES en el contexto regional NEA.

- Diagnóstico de la importancia que las Pymes de software confieren a los modelos y métodos de aseguramiento de la calidad, así como a las herramientas utilizadas para tal fin.
- Metodología que permita evaluar la calidad en uso de productos software sobre plataformas Web y que brinde información que aporte a la mejora del proceso de creación de software.
- Nuevas herramientas o adaptación de las existentes para automatizar la generación de los indicadores que conforman las métricas de evaluación de la calidad en uso de aplicaciones web.
- Vinculaciones con otras instituciones, universitarias, gubernamentales o empresariales (especialmente las PYMES regionales), para el estudio y desarrollo de metodologías y/o desarrollo de herramientas relacionadas con la evaluación de la calidad en el software.-
- Especialización de recursos humanos en los temas propios de la evaluación de calidad de software a través de la elaboración de tesis de carreras de posgrado y trabajos de investigación realizados en el marco de becas de investigación para alumnos y/o graduados.
- Realización de transferencias tecnológicas o de capacitación en temas relacionados con la mejora de la calidad de productos software desarrollados por Pymes regionales/nacionales.

La implementación de este proyecto permitirá el desarrollo de actividades de investigación en un área actualmente considerada de vacancia en la región de influencia de la UNNE y la UTN - Facultad Regional Resistencia, posibilitando la inserción y el crecimiento en el camino de la investigación de alumnos y recientes graduados de las carreras de Informática ofrecidas por ambas universidades.

A su vez, contribuirá a mejorar la formación de los profesionales de la Informática dado que la generación y la actualización de los conocimientos, se trasladará a los alumnos a través del proceso de enseñanza y aprendizaje, en cada una de las asignaturas dictadas por los integrantes del proyecto. Finalmente, permitirá que las universidades realicen transferencia de conocimientos al medio regional, contribuyendo de esta forma a mejorar la competitividad de las empresas Pymes, generar valor para la zona mediante la creación de puestos de trabajo, facilitar la inserción de los profesionales formados en la universidad, evitando el desarraigo de los jóvenes graduados y la migración a los grandes centros urbanos, cambios importantes que conllevan un deterioro de la calidad de vida de las personas.

#### 4. FORMACION DE RECURSOS HUMANOS

En el marco del proyecto se encuentran en este momento en desarrollo un plan de trabajo de beca de investigación de pregrado de la SECYT-UNNE, se proyecta el desarrollo de cinco tesis de la Maestría en Ingeniería de Software de la UNLP dictada en FaCENA-UNNE y UTN-FRRe, mediante acuerdo institucional, como así también, la elaboración de dos tesis doctorales, correspondientes al Doctorado en Ingeniería de Sistemas y Computación de la Universidad de Málaga- España.

#### 5. BIBLIOGRAFIA

- [1] Primer Foro Nacional de experiencias en el Sector TIC. "El Encuentro de los Polos de Argentina". 2007. Disponible en: [http://www.ses.me.gov.ar/spu/Noticias/Noticias\\_Universitarias\\_2007/noticias\\_marzo\\_2007\\_asociacion\\_de\\_tec.htm](http://www.ses.me.gov.ar/spu/Noticias/Noticias_Universitarias_2007/noticias_marzo_2007_asociacion_de_tec.htm)
- [2] Pressman, R. S. "Ingeniería de Software. Un enfoque práctico". Editorial MCGRAW-HILL - 2005
- [3] McCall, J.A. ; Richards, P.K. ; Walters, G.F. – "Factors in Software Quality". Vols I, II, III. NTIS AD-AO49-014, 015, 055, Nov. 1977
- [4] AENOR. UNE-EN-ISO 8402 "Gestión de la calidad y aseguramiento de calidad". Vocabulario (ISO 8402:1994).
- [5] IEEE Standard 610. Institute of Electrical and Electronics Engineers Computer dictionary. Compilation of IEEE Standard Computer Glossaries. 1990
- [6] Ian Sommerville. "Ingeniería del Software". 7ª Ed. Cap. 27. Pearson Educacion S.A., Madrid 2005.
- [7] Scalone F. "Estudio comparativo de los modelos y estándares de calidad del software". Tesis de Maestría en Ingeniería en Calidad. Universidad de Buenos Aires. 2006. Disponible en <http://www.fi.uba.ar/laboratorios/lisi/scalone-tesis-maestría-ingeniería-en-calidad.pdf>
- [8] Piattini, M., García F., Caballero, I. "Calidad de los Sistemas Informáticos". Editorial Alfaomega. 2007.
- [9] Marín B., Condori-Fernández N., Pastor O., "Calidad en Modelos Conceptuales: Un Análisis Multidimensional de Modelos Cuantitativos basados en la ISO 9126". Revista de Procesos y Métricas.
- [10] ISO, "ISO/IEC 9126-1 – Software engineering – Product quality – Part 1: Quality Model", 2001.
- [11] ISO, "ISO/IEC 9126-2 – Software engineering – Product quality – Part 2: External Metrics", 2003.
- [12] ISO, "ISO/IEC 9126-3 – Software engineering – Product quality – Part 3: Internal Metrics", 2003.
- [13] ISO, "ISO/IEC 9126-4 – Software engineering – Product quality – Part 4: Quality in Use Metrics", 2004.
- [14] Ruiz Morilla, J., "ISO 9126 vs. SQuARE". Material del curso de posgrado Calidad y Medición de Sistemas de Información. Escuela Superior de Informática. Universidad de Castilla-La Mancha.

2008. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Joaquin%20Ruiz%20E.xpo.pdf>

[15] Gómez García, O. "SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad". Material del curso de posgrado Calidad y Medición de Sistemas de Información. Escuela Superior de Informática. Universidad de Castilla-La Mancha. 2009. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Oscar%20Gomez.pdf>

[16] Olsina L. et al "Specifying Quality Characteristics and attributes for web sites". ACM. 1999.

[17] Offutt, J. "Quality attributes of web software applications", in *IEEE Software*. 2002, pp.25-32

[18] Rodríguez, D., Harrison, R., "Medición en la Orientación a Objeto", in *Medición para la Gestión en la Ingeniería del Software*, Dolado, J. and Fernández, L., Eds., RA-MA, 2000, ISBN 84-7897-403-2.

[19] Marchetto, A. "A concerns-based metrics suite for web applications". In *INFOCOMP Journal of Computer Science*, 4(3):11–22, 2005.

[20] Grigoreta, M. S.; Serban, G. - "Quality measures for evaluating the results of clustering based aspect mining techniques". In *Proc. of TEAM '06 (Towards Evaluation of Aspect Mining) Workshop in ECOOP '06*, pages 13–16. TU Delft, June 2006.

[21] Zhao, J. - "Towards a Metric Suite for Aspect-Oriented Software". - Technical Report SE-136-25, Information Processing Society of Japan (IPSJ), Marzo 2002.

[22] Chicano García, Francisco. "Metaheurísticas e Ingeniería del Software". Phd Thesis. Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga. España. 2007.

[23] Des Greer and Günther Ruhe. "Software release planning: an evolutionary and iterative approach." En *Information and Software Technology*, 46(4), 243:253, 2004.

[24] C. L. Simons and I. C. Parmee. "Single and multi-objective genetic operators in object-oriented conceptual software design". In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1957:1958. ACM Press, 2006.

[25] Andy Nisbet. "GAPS: A compiler framework for genetic algorithm (GA) optimised parallelisation". In *High-Performance Computing and Networking, International Conference and Exhibition, HPCN Europe 1998, Amsterdam, The Netherlands, April 21-23, 1998, roceedings*, volume LNCS 1401, pages 987:989. Springer, 1998

[26] Kenneth Peter Williams. "Evolutionary Algorithms for Automatic Parallelization". PhD thesis, University of Reading, UK, 1998.

[27] Enrique Alba y Francisco Chicano. "Software project management with Gas". *Information Sciences*, 177(11), 2380:2401, Junio 2007.

[28] Enrique Alba y J. Francisco Chicano. "Software testing with evolutionary strategies". En *Proceedings of the 2nd International Workshop on Rapid Integration of Software Engineering Techniques, LNCS 3943*, páginas 50:65, Heraklion, Grecia, Septiembre 2005.