

**ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO
NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE
TRATAN LA DISCIPLINA**

**ALONSO TORO LAZO
LORENA CARDONA BENJUMEA
Asesor: Ing. LUIS EDUARDO PELÁEZ VALENCIA**



**UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES**

PEREIRA, 2010

DEDICATORIA

A nuestras familias y en especial a la memoria de nuestros padres *José Darío Toro Restrepo* y *José Yeiro Cardona Osorio*.

AGRADECIMIENTOS

A Dios

Por guiar e iluminar nuestro camino

A nuestros padres

Por su amor, comprensión y apoyo incondicional

A nuestro@s hermano@s

Por alentarnos a alcanzar nuestras metas

A nuestro@s amigo@s

Por su compañía y aliento en cada momento

A nuestros mentores y en especial a nuestro asesor

Por su orientación y aporte a nuestra formación

TABLA DE CONTENIDO

	Página
RESUMEN	1
INTRODUCCIÓN	2
1. JUSTIFICACIÓN.....	4
2. PLANTEAMIENTO DEL PROBLEMA.....	6
3. DEFINICIÓN DEL PROBLEMA.....	7
4. OBJETIVOS	8
4.1 Objetivo general.....	8
4.2 Objetivos específicos	8
5. ALCANCES.....	9
6. METODOLOGÍA	10
7. CONTEXTUALIZACIÓN	11
7.1 Historia de la ingeniería del software	11
7.2 Organizaciones relacionadas con la ingeniería del software a nivel internacional.....	12
7.2.1 IEEE (The Institute of Electrical and Electronics Engineers, inc).....	12
7.2.2 ISO (Organización Internacional de Normalización).....	16
7.2.3 Comisión Electrotécnica Internacional (IEC)	17
7.2.4 European Software Institute (ESI)	18
7.2.5 NASSCOM (India)	19

7.2.6 American National Standards Institute (ANSI).....	20
7.2.7 Comisión Panamericana de Normas Técnicas (COPANT)	20
7.2.8 International Software Benchmarking Standards Group (ISBSG)	24
7.2.9 Centre for Software Engineering (CSE).....	25
7.2.10 Association for Computing Machinery (ACM).....	25
7.2.11 European Space Agent (ESA).....	27
7.2.12 British Standards Institution (BSI).....	28
7.2.13 Red sobre Experimentación y Medición en Ingeniería de Software (REMIS).....	29
7.2.14 Otros organismos de normalización a manera de información.....	30
7.3 Organizaciones que tienen que ver con el desarrollo de software en Latinoamérica	36
7.3.1 CESSI (Argentina).....	36
7.3.2 SOFTEX (Brasil).....	37
7.3.3 GECHS (Chile)	39
7.3.4 CAMTIC (Costa Rica).....	40
7.3.5 AESOFT (Ecuador)	41
7.3.6 SOFEX (Guatemala)	41
7.3.7 APESOFT (Perú).....	42
7.3.8 AMITI (México)	43
7.3.9 CAVEDATOS (Venezuela)	44
7.3.10 CUTI (Uruguay)	44
7.3.11 CTIP (Paraguay).....	45
7.3.12 CBTI (Bolivia)	45
7.3.13 CONACYT (El Salvador)	46

7.4 Organizaciones que tienen que ver con el desarrollo de software en Colombia.....	46
7.4.1 FEDESOFTE (Colombia).....	46
7.5 Conceptualización	47
7.5.1 Norma.....	47
7.5.2 Estándar	48
7.5.3 Regulación	48
7.5.4 Modelo de Procesos de Software.....	49
7.5.5 Metodología.....	49
7.5.6 Guía.....	51
8. PRESENTACIÓN Y ANÁLISIS INDIVIDUALIZADO DE RESULTADOS	52
8.1 Estándares y normas.....	52
8.2 Guías	70
8.3 Modelos de Desarrollo de Software.....	72
8.3.1 Modelos Clásicos	72
8.3.2 Modelos Evolutivos del Proceso de Software.....	77
8.3.3 Desarrollo Basado en Componentes.....	82
8.3.4 El Modelo de Métodos Formales	84
8.3.5 Modelos de Calidad del Software.....	85
8.4 Metodologías	93
8.4.1 Metodologías Estructuradas.....	98
8.4.2 Metodologías Orientadas a Objetos	114
8.4.3 Metodologías de Desarrollo Ágil.....	126
8.5 Propuestas para el aseguramiento de la calidad del software en Colombia. 143	
8.5.1 PRATI.....	146

8.5.2	COMPETISOFT.....	146
8.5.3	SIMEP-SW	147
8.6	Herramientas	149
8.6.1	CASE.....	149
8.6.2	UML (Unified Modeling language).....	150
9.	CONCLUSIONES Y RECOMENDACIONES	153
10.	BIBLIOGRAFÍA	155
11.	WEBGRAFÍA.....	156

INDICE DE TABLAS

	Página
Tabla 1: Organismos Internacionales de Normalización	30
Tabla 2: Organismos de Normalización en América	32
Tabla 3: Organismos de Normalización en África	33
Tabla 4: Organismos de Normalización en Asia.....	34
Tabla 5: Organismos de Normalización en Europa	36
Tabla 6: Organismos de Normalización en Oceanía	36
Tabla 7: Análisis de la Norma ISO/IEC 9126-1 (2001)	54
Tabla 8: Análisis de la Norma ISO/IEC TR 9126-2 (2003)	55
Tabla 9: Análisis de la Norma ISO/IEC TR 9126-3 (2003)	57
Tabla 10: Análisis de la Norma ISO/IEC TR 9126-4 (2004)	58
Tabla 11: Análisis del Estándar ISO/IEC 14598 (1998-2001)	59
Tabla 12: Análisis de la norma ISO 25000 (2005).....	61
Tabla 13: Análisis de la Norma ISO/TR 9241-100 (2010)	63
Tabla 14: Análisis de la Norma ISO 9241-171 (2008)	64
Tabla 15: Análisis del estándar ISO/IEC 90003 (2004)	66
Tabla 16: Análisis de la norma ISO/IEC 12207 (2008).....	68
Tabla 17: Análisis del estándar IEEE 1058.1 (1987)	70
Tabla 18: Análisis de la Guía SWEBOK (2004).....	71
Tabla 19: Análisis del Modelo Lineal, Secuencial o Cascada	73
Tabla 20: Análisis del Modelo de Construcción de Prototipos.....	75
Tabla 21: Análisis del Modelo DRA (Desarrollo Rápido de Aplicaciones)	77
Tabla 22: Análisis del Modelo Incremental	78
Tabla 23: Análisis del Modelo en Espiral.....	80
Tabla 24: Análisis Modelo Espiral WIN-WIN	81
Tabla 25: Análisis del Modelo de Desarrollo Concurrente.....	82
Tabla 26: Análisis del Modelo de Desarrollo Basado en Componentes	83

Tabla 27: Análisis del Modelo de Métodos Formales	85
Tabla 28: Análisis del Modelo CMM (Capability Maturity Model).....	86
Tabla 29: Análisis del Modelo CMMI Versión 1.2 (2006).....	88
Tabla 30: Análisis del Modelo de Procesos para la Industria del Software (MoProSoft)	89
Tabla 31: Análisis del Modelo MPS (Mejora de Proceso del Software).....	91
Tabla 32: Análisis del Modelo SPICE (1993).....	92
Tabla 33: Características generales de las metodologías estructuradas y orientadas a objetos.....	95
Tabla 34: Metodologías estructuras y orientadas a objetos	95
Tabla 35: Convergencias y divergencias entre las principales metodologías ágiles .	96
Tabla 36: Metodologías más representativas en la historia de la Ingeniería del Software	97
Tabla 37: Análisis de la Metodología MERISE (1978).....	99
Tabla 38: Análisis de la Metodología SSADM Versión inicial (1981).....	100
Tabla 39: Análisis de la Metodología SSADM Versión 3 (1986).....	100
Tabla 40: Análisis de la Metodología SSADM Versión 4: 1990	103
Tabla 41: Análisis de la Metodología MÉTRICA Versión 2 (1993)	104
Tabla 42: Análisis de la Metodología MÉTRICA Versión inicial (1989)	104
Tabla 43: Análisis de la Metodología MÉTRICA Versión 2.1 (1995)	105
Tabla 44: Análisis de la Metodología PRINCE1: 1989	108
Tabla 45: Análisis de Metodología MÉTRICA Versión 3 (2001).....	108
Tabla 46: Análisis de la Metodología PRINCE2 (2005).....	111
Tabla 47: Análisis de la Metodología MAGERIT Versión 2.0 (2006).....	112
Tabla 48: Análisis de la Metodología MAGERIT Versión 1.0	112
Tabla 49: Análisis de la Metodología EUROMÉTODO Versión 1 (1998)	114
Tabla 50: Análisis de la Metodología RUP (1998).....	118
Tabla 51: Análisis de la Metodología OORAM (1996).....	121
Tabla 52: Análisis de la Metodología OMT (1991)	123
Tabla 53: Análisis de la Metodología SEOO	124

Tabla 54: Análisis de la Metodología OOAD (1991).....	124
Tabla 55: Otras Metodologías Orientadas a Objetos	126
Tabla 56: Análisis de la Metodología XP - eXtreme Programming (1991)	130
Tabla 57: Análisis de la Metodología Scrum (1995)	133
Tabla 58: Análisis de la Familia Metodológica Crystal (1998)	135
Tabla 59: Análisis de la Metodología DSDM (1994).....	137
Tabla 60: Análisis de la Metodología DFF (1998 y 2002).....	140
Tabla 61: Análisis de la Metodología MSF Versión 4.0 (2005).....	143

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

RESUMEN

En la actualidad, la calidad es un aspecto clave en el desarrollo e implementación de soluciones a través del software. Es por este motivo, que diferentes organizaciones han generado diversas prácticas de ingeniería para garantizarla. El presente trabajo de grado aborda de manera general el estado del arte de la ingeniería del software en el ámbito nacional e internacional de acuerdo a organizaciones que tratan la disciplina, con el fin de servir como referente para nuevas propuestas que permitan abordar la ingeniería del software como disciplina, a nivel local, nacional o internacional; de tal manera que no se desconozca el avance que, desde los años 50, se ha logrado al respecto.

Palabras Clave: Ingeniería del software, calidad, estado del arte.

ABSTRACT

Nowadays, quality is a key aspect in the development and implementation of solutions through software. For this reason, different organizations have created several engineering practices to ensure it. This project shows broadly the state of the art of software engineering according to national and international organizations that deal with this discipline, to serve as reference for new proposals to address the software engineering as a discipline at the local, national or international level so that it recognizes the progress that since the 50's, has been achieved in this regard.

Key words: Software engineering, quality, state of art.

INTRODUCCIÓN

La **Ingeniería del Software** es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos o desarrollos en Intranet/internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet, redes Intranet y Extranet, etc. (S. Pressman, 2002, pág. XXIX)

Existen diversas metodologías y estándares creados por diferentes organismos internacionales de gran prestigio como la IEEE y la ISO para mediar y certificar la calidad en la creación, proceso de desarrollo y evaluación de productos software. Sin embargo, muchos países no tienen una identidad en cuanto al uso de políticas de desarrollo¹ y han optado por la implementación de diversas técnicas de origen extranjero para el desarrollo de sus aplicaciones como es el caso de Colombia.

Debido a la ausencia de estándares propios, muchos desarrolladores no tienen en cuenta los procesos de la ingeniería del software para construir sus aplicaciones, ya

¹ En Colombia, según estudios como el realizado por DATANALISIS en el año 2005, citado en el documento Planificación Programas de Apoyo a las Tecnologías de Información (PRATI), Ministerio de comunicaciones, 2008, se indica que la falta de una política de estado para la industria del software corresponde al 31.2% de los principales problemas que enfrenta la industria de software nacional, así como el no contar con un marco normativo adecuado representa el 24.7%.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

que éstos revisten alguna complejidad, y por el contrario, se limitan a programar sin tener unos criterios previamente establecidos, dando como resultado software de baja calidad.

Éste documento presenta un estudio en el que se pretende dar una idea del estado del arte de la ingeniería del software en el ámbito nacional e internacional a partir de organizaciones que tratan la disciplina para que, en un futuro, se puedan definir estándares propios, que permitan la construcción y el mantenimiento de aplicaciones de excelente calidad.

La investigación se aborda a partir de las organizaciones, porque se considera que el estado del arte en la ingeniería del software se puede determinar mejor a partir de publicaciones realizadas por periódicos sobre la industria, libros de diferentes autores reconocidos, series monográficas de profesionales académicos tanto de universidades como de la industria y del gobierno, así como una gran variedad de estándares y procedimientos de ingeniería del software desarrollados y producidos por organizaciones reconocidas internacionalmente como la IEEE (*Institute of Electrical and Electronics Engineers*), ISO (International Organization for Standardization) y ESA (European Space Agency), además de artículos promocionados por organizaciones como ACM (Association for Computing Machinery), autoridades civiles de aviación y otras agencias del gobierno y sin ánimo de lucro que proporcionan una referencia detallada de los estándares de ingeniería del software desarrollados más recientemente.

1. JUSTIFICACIÓN

Con la gran expansión de las tecnologías de la información y la comunicación, se ha incrementado considerablemente la producción de equipos que permiten la sistematización de los procesos realizados por el hombre en diferentes campos. A la par de estos avances también se ha incrementado el desarrollo de software, y con el pasar de los años, la Ingeniería del Software ha introducido y popularizado una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar como del proceso de desarrollo en sí.

En la actualidad, la economía mundial depende más de sistemas automatizados que en otras épocas, ésta situación ha provocado que los equipos de desarrollo tengan que enfrentarse día a día con nuevos procesos y estándares de calidad. Para afrontar éste inconveniente, se han desarrollado varias herramientas automatizadas que permiten definir y aplicar procesos de desarrollo de software más efectivos.

Así pues, se hace evidente que la ingeniería del software se ha convertido en un elemento fundamental para garantizar que la construcción y el mantenimiento de productos de software de alta calidad se haga de manera eficiente, siendo muy importante que el desarrollador, o quien haga las veces de constructor de software, tenga conocimiento de las metodologías que se utilizan en diferentes partes del mundo para desarrollar éste tipo de productos, con el fin de utilizar las experiencias encontradas en la creación de nuevas aplicaciones.

Teniendo en cuenta lo anterior, se puede decir que indagar acerca del estado del arte de la ingeniería del software en el ámbito nacional e internacional, es un

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E
INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ejercicio útil para la generación de políticas y/o estándares regionales de creación de software, de igual forma, se convierte en punto de partida para aquellos que buscan nuevas alternativas de desarrollo.

2. PLANTEAMIENTO DEL PROBLEMA

Desde su aparición, el software ha sido utilizado en una extensa variedad de áreas de aplicación, razón por la cual se ha convertido en una herramienta fundamental e indispensable para el crecimiento de la sociedad a nivel económico, industrial, tecnológico y cultural, entre otros. Su inclusión en la vida del hombre, ha provocado que la realización de sus labores diarias dependa en gran medida de la utilización de éste tipo de aplicaciones.

Es por este motivo, que para desarrollar un software con éxito, que funcione impecablemente, que sea fácil de utilizar y sobre todo, que satisfaga las necesidades de las personas que lo utilizan, es necesario tener en cuenta los lineamientos de la ingeniería del software con el fin de garantizar su calidad y con ello el mejoramiento continuo de los procesos. Sin embargo, en la actualidad muchos países no tienen una identidad en cuanto al uso de políticas de desarrollo y han optado por la implementación de diversas técnicas de origen extranjero para producir sus aplicaciones, como es el caso de Colombia, nación que a pesar de haber hecho esfuerzos por mejorar su procesos de ingeniería de software, no ha logrado definir un estándar propio que le permita a sus desarrolladores tener una base para llevar a cabo la creación y certificación de sus productos.

3. DEFINICIÓN DEL PROBLEMA

A manera de hipótesis se ha considerado que el software producido en Colombia es de baja calidad, dado que el país no cuenta con un estándar propio para la creación, evaluación, certificación y mantenimiento del mismo. Así pues, se pretende realizar el estudio exploratorio mencionado anteriormente, de tal manera, que permita identificar las falencias y fortalezas de las diferentes técnicas existentes, facilitando la creación de un estándar para Colombia que defina una serie de políticas y/o procedimientos que se puedan llevar a cabo en la construcción de software, que satisfaga las necesidades de las organizaciones y usuarios finales.

Tener software de baja calidad redundará en procesos con deficiencias al interior de las organizaciones y por ende en un inferior nivel competitivo nacional respecto a los países latinoamericanos, también en vía de desarrollo; lo que afectaría, inclusive la economía del país, en la tendencia de la firma de acuerdos de libre comercio que permitirían que los países que mejor preparados se encuentren en fabricación de software de calidad, reduzcan las posibilidades de comercialización del producto nacional.

4. OBJETIVOS

4.1 Objetivo general

Explorar y documentar el estado del arte de la Ingeniería del Software en el ámbito nacional e internacional de acuerdo a organizaciones que tratan la disciplina.

4.2 Objetivos específicos

- Diagnosticar el estado del arte de la ingeniería del software en Colombia a partir de la documentación existente disponible.
- Diagnosticar el estado del arte de la ingeniería del software a nivel internacional de acuerdo a organizaciones que tratan la disciplina.
- Elaborar el documento final del estado del arte de la ingeniería del software en el ámbito nacional e internacional de acuerdo a organizaciones que tratan la disciplina.

5. ALCANCES

Se pretende realizar una investigación exploratoria a cerca del estado del arte de la ingeniería del software en el ámbito nacional e internacional a partir de organizaciones que tratan la disciplina, que facilite en un futuro la definición de un estándar para la construcción y el mantenimiento de software de excelente calidad en Colombia. La exploración contempla la información considerada por las bases de datos especializadas del orden académico que puedan ser consultadas desde la Universidad Católica Popular del Risaralda, la información que comparten las asociaciones que agremian el sector del desarrollo del software, las organizaciones internacionales que puedan ser consultadas a través de internet y los libros que tratan la disciplina a través de modelos y/o metodologías según se expresa en el capítulo siguiente.

6. METODOLOGÍA

La metodología a utilizar consiste en realizar un estudio exploratorio sobre el uso de la ingeniería del software a nivel nacional e internacional, contemplando dentro de éste las organizaciones, los estándares, las métricas, las técnicas y demás prácticas utilizadas en la producción de software de calidad. Lo anterior, teniendo en cuenta que éste es un tema sobre el cual hay diferentes posturas y por ende, la adopción de un procedimiento determinado o una metodología específica depende del contexto en el que se encuentre el desarrollador. Por lo anterior, en la primera parte del estudio se pretende obtener una visión general acerca del concepto a través de un recorrido teórico y el estado de las prácticas mediante su contextualización en el ámbito internacional. Posteriormente se realizará un análisis del entorno Colombiano, ya que la ingeniería del software no ha sido un tema ampliamente estudiado, razón por la cual el país no cuenta actualmente con un estándar propio para producir software de calidad.

7. CONTEXTUALIZACIÓN

7.1 Historia de la ingeniería del software

El término *ingeniería del software* empezó a usarse a finales de la década de los sesenta, para expresar el área de conocimiento que se estaba desarrollando en torno a las problemáticas que ofrecía el software en ese momento. En esa época, el crecimiento espectacular de la demanda de sistemas de computación cada vez más y más complejos, asociado a la inmadurez del propio sector informático (totalmente ligado al electrónico) y a la falta de métodos y recursos, provocó lo que se llamó *la crisis del software* (en palabras de Edsger Dijkstra) entre los años 1965 y 1985. Durante esa época muchos proyectos importantes superaban con creces los presupuestos y fechas estimados, algunos de ellos eran tan críticos (sistemas de control de aeropuertos, equipos para medicina, etc.) que sus implicaciones iban más allá de las pérdidas millonarias que causaban.

Desde 1985 hasta el presente, han ido apareciendo herramientas, metodologías y tecnologías que se presentaban como la solución definitiva al problema de la planificación, previsión de costes y aseguramiento de la calidad en el desarrollo de software (Gilbert & Peña, 2005), sin embargo éstas no siempre son aplicadas adecuadamente ya que los desarrolladores aún no se han apropiado de la ingeniería del software como una disciplina que les permita dar soluciones efectivas a las problemáticas planteadas. También es importante resaltar que cada una de las metodologías existentes en la actualidad, han sido producto de unas necesidades específicas que se han ido presentando en diferentes partes del mundo, es por ésta razón que no se cuenta con un estándar único que se pueda aplicar a todos los procesos de desarrollo de software.

7.2 Organizaciones relacionadas con la ingeniería del software a nivel internacional

Debido a las diferentes problemáticas presentadas a nivel internacional en cuanto al desarrollo de software, muchas organizaciones se han unido a la búsqueda de nuevas alternativas que fomenten las buenas prácticas de la ingeniería del software con el fin de garantizar la producción de soluciones que satisfagan los requerimientos de los clientes. Algunas de ellas son:

7.2.1 IEEE (The Institute of Electrical and Electronics Engineers, inc)

Creado en Nueva York en 1884, es una asociación internacional sin ánimo de lucro con sede principal en la ciudad de Piscataway en los Estados Unidos y subseces en más de 190 países del mundo, con alrededor de 370.000 miembros, entre profesionales y estudiantes de ingeniería, diseño, derecho, administración, medicina, biología y ciencias afines.

Su misión es fomentar la prosperidad global para beneficio de la humanidad y las profesiones, mediante la promoción de los procesos de ingeniería, en la creación, desarrollo, integración, participación y aplicación del conocimiento de la informática, la ciencia electromagnética y la electrotecnología.

Organización interna:

El Instituto está conformado por una división técnica y otra geográfica. La división técnica está compuesta por 39 diferentes sociedades, cada una de ellas orientada a un tema macro del conocimiento tecnológico. La división geográfica comprende 10 regiones a nivel mundial, compuestas a su vez por 329 secciones locales. Colombia pertenece al IEEE Región 9 (Latino América) y a la Sección Colombia. Además, se tienen diferentes tipos de membresía, clasificados básicamente en dos grandes grupos: Miembros Profesionales y Miembros Estudiantiles.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

IEEE cuenta con 1.784 capítulos integrados por miembros locales con similares intereses técnicos, 38 sociedades y 7 consejos técnicos, 1.616 ramas estudiantiles en colegios y universidades en 80 países, y 452 capítulos en la rama estudiantil.

Su misión es fomentar la prosperidad global para beneficio de la humanidad y las profesiones, mediante la promoción de los procesos de ingeniería, en la creación, desarrollo, integración, participación y aplicación del conocimiento de la informática, la ciencia electromagnética y la electrotecnología.

Miembros:

Hay más de 365.000 miembros del IEEE en más de 160 países de todo el mundo, entre los que se encuentran ingenieros, científicos y profesionales cuyos intereses técnicos se basan en eléctrica y ciencias de la computación, ingeniería y disciplinas afines.

Estándares:

El IEEE es un desarrollador líder de las normas internacionales que sustentan muchas de las telecomunicaciones de hoy en día, la tecnología de la información y los productos de la generación de energía y servicios. Es a menudo la fuente principal para la normalización en una amplia gama de tecnologías emergentes. Desarrolla sus estándares a través de una de sus entidades, la IEEE Standards Association (IEEE-SA). Asimismo, este desarrollo se potencia mediante otras entidades técnicas abarcadas por el Instituto, la IEEE Computer Society (IEEE-CS) y el IEEE Technical Council on Software Engineering (TCSE), las cuales participan de esta actividad mediante un comité, el Software & Systems Engineering Standards Committee (S2ESC).

7.2.1.1 IEEE Standards association (IEEE-SA).

La IEEE Standards Association (IEEE-SA) produce las normas que respondan a las necesidades globales de la industria, el gobierno y el público para una amplia gama de tecnologías e industrias. Cuenta con una cartera de alrededor de 900 normas activas y más de 400 normas en preparación. Tiene relaciones estratégicas con el IEC², la ISO³, la UIT⁴ y SDO⁵.

7.2.1.2 IEEE Computer society.

Computer Society es la más grande de las sociedades organizadas en el IEEE y el organismo líder en proveer información técnica y servicios a estudiantes y profesionales de la computación, la informática y los sistemas a nivel mundial.

Con cerca de 85.000 miembros, la IEEE Computer Society es la organización líder en el mundo de profesionales de la informática. Fundada en 1946, y la mayor de las 38 sociedades del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), se dedica

² <http://www.iec.ch>. International Electrotechnical Commission. Organización líder en el mundo que prepara y publica estándares internacionales para todas las tecnologías eléctricas, electrónicas y relacionadas - conocidos colectivamente como "electrotecnia".

³ <http://www.iso.org>. International Organization for Standardization. Es el desarrollador más grande del mundo y editor de las Normas Internacionales.

⁴ <http://www.itu.int>. Unión Internacional de Telecomunicaciones. Es la organización más importante de las Naciones Unidas en lo que concierne a las tecnologías de la información y la comunicación.

⁵ <http://sdo.gsfc.nasa.gov>. Solar Dynamics Observatory. Es la primera misión que se lanzará para la NASA Living With a Star (LWS) Program, un programa diseñado para entender las causas de la variabilidad solar y sus efectos en la Tierra.

a avanzar la teoría y aplicación de la informática y tecnología de la información de procesamiento.

En el año 2004 la IEEE Computer Society publicó la Guía SWEBOK, en la cual se establece por primera vez una línea de base para el cuerpo de conocimiento en el ámbito de la ingeniería de software, cumpliendo parcialmente con la responsabilidad de promover el adelanto de la teoría y la práctica en este campo. Cabe señalar que la Guía no pretende definir el conjunto de conocimientos, sino más bien servir como un compendio y guía para el desarrollo y la evolución de los mismos.

7.2.1.3 The IEEE Technical Council on Software Engineering (TCSE).

El Consejo Técnico IEEE de Ingeniería del Software (TCSE) alienta a la aplicación de métodos de ingeniería y principios para el desarrollo de programas de computadora, y trabaja para incrementar los conocimientos profesionales de las técnicas, herramientas y datos empíricos para mejorar la calidad del software. TCSE participa en las múltiples formas en que el software es diseñado, desarrollado, administrado y mantenido.

7.2.1.4 Software & Systems Engineering Standards Committee (S2ESC).

Es el Comité de normas para software e ingeniería de sistemas del IEEE cuya misión es desarrollar y mantener una familia de software y sistemas de normas de ingeniería que sea pertinente, coherente, completa y eficaz. Estas normas son implementadas por profesionales, organizaciones y educadores con el fin de mejorar la eficacia y eficiencia de los procesos de ingeniería de software, la comunicación entre compradores y proveedores, y la calidad del software entregado.

7.2.2 ISO (Organización Internacional de Normalización).

ISO es el desarrollador más grande del mundo y editor de las Normas Internacionales. ISO es una red de los institutos nacionales de normalización de 163 países, un miembro por país, con una Secretaría Central en Ginebra, Suiza, que coordina el sistema.

La ISO es una organización no gubernamental que forma un puente entre los sectores público y privado. Por un lado, muchos de sus institutos miembros son parte de la estructura gubernamental de sus países, o están obligados por su gobierno. Por otra parte, otros miembros tienen sus raíces únicamente en el sector privado, después de haber sido creado por las sociedades nacionales de las asociaciones de la industria.

Por lo tanto, la norma ISO permite llegar a un consenso sobre las soluciones que satisfagan tanto los requisitos de negocio y las necesidades más amplias de la sociedad. ISO cuenta con más de 17500 Normas Internacionales y otros tipos de documentos normativos en su cartera actual.

7.2.2.1 Orígenes de la ISO.

ISO nació de la unión de dos organizaciones - la ISA (Federación Internacional de las Asociaciones de Normalización Nacional), establecida en Nueva York en 1926, y el UNSCC (Naciones Unidas Standards Coordinating Committee), establecida en 1944.

En octubre de 1946, delegados de 25 países, reunidos en el Instituto de Ingenieros Civiles de Londres, decidió crear una nueva organización internacional, cuyo objeto sería "facilitar la coordinación internacional y la unificación de las normas industriales". La nueva organización, ISO, inició oficialmente sus operaciones el 23 de febrero de 1947.

ISO es la mayor organización del mundo en desarrollo de normas. Entre 1947 y el día de hoy, ha publicado más de 17.500 normas internacionales, que van desde las normas para actividades como la agricultura y la construcción, la ingeniería mecánica, los productos sanitarios hasta la evolución más reciente de la tecnología de la información.

7.2.3 Comisión Electrotécnica Internacional (IEC)

La Comisión Electrotécnica Internacional (IEC) es la organización líder a nivel mundial encargada de preparar y publicar Normas Internacionales para todas las tecnologías eléctricas, electrónicas y afines.

Las actividades de la IEC - conocidas en su conjunto como "electrotecnología" - cubren todas las tecnologías eléctricas, electrónicas y afines en el ámbito terrestre, marítimo y aéreo, además de disciplinas tales como la terminología, la compatibilidad electromagnética, el estudio del funcionamiento, la seguridad y el medioambiente, así como la optimización de la eficiencia de la energía eléctrica y el desarrollo de normas para las energías renovables.

A la vez que constituyen un excelente marco de trabajo para mejorar los parámetros de seguridad y optimizar el uso de energía, las Normas Internacionales IEC facilitan el comercio entre países al proporcionar una referencia para el funcionamiento del Acuerdo sobre Obstáculos Técnicos al Comercio (OTC) de la Organización Mundial del Comercio (OMC).

A los Miembros IEC se les conoce como "Comités Nacionales" y existe uno por país. Cada Comité Nacional representa plenamente a todas las partes interesadas en el campo de la electrotecnología a nivel nacional. Generalmente los Comités Nacionales están constituidos por representantes de la industria, los organismos

gubernamentales, el mundo académico, las asociaciones comerciales, los usuarios y los creadores de normas. Los Comités Nacionales de la IEC están conformados de diversas maneras. Algunos sólo representan al sector público, algunos presentan una combinación del sector público y privado, y otros sólo representan al sector privado.

Los Comités Técnicos (CTs) o Subcomités (SCs) preparan Normas Internacionales para un área específica de la electrotecnología. Los productos o publicaciones resultantes de éste trabajo se distribuyen en dos amplias categorías: Normativa e Informativa. Las publicaciones normativas pueden ser Normas Internacionales, Especificaciones Técnicas, Especificaciones Disponibles al Público y Acuerdos Técnicos de la Industria. Las publicaciones informativas pueden ser Informes Técnicos, Evaluaciones de Tendencias Tecnológicas y Guías.

El uso de Normas Internacionales IEC para la certificación a nivel nacional garantiza que un producto certificado ha sido fabricado y evaluado según criterios rigurosos y bien establecidos. El usuario final puede estar seguro de que el producto en cuestión satisface las normas mínimas de calidad y no necesita pasar más evaluaciones o pruebas.

7.2.4 European Software Institute (ESI)

El European Software Institute (ESI), es una organización privada sin fines de lucro. Centro Tecnológico lanzado como una iniciativa de la Comisión Europea, con el apoyo del Gobierno Vasco y las principales empresas europeas que trabajan en el campo de la Tecnología de la Información.

La principal actividad de ESI se basa en ayudar a la industria del software en sus objetivos de producir mejor software de mayor calidad, a tiempo, de la mejor manera y con un costo menor. ESI ofrece servicios de consultoría y formación, así como el

apoyo tecnológico a través de su equipo de expertos altamente cualificados y una red de alianza en todo el mundo.

ESI desarrolla también una amplia gama de iniciativas destinadas a promover el desarrollo de la Sociedad de la Información a través del uso de las TIC para los ciudadanos y las empresas, especialmente las PYMES y las microempresas.

7.2.5 NASSCOM (India)

NASSCOM es el principal organismo de comercio y de la Cámara de Comercio de la industria IT-BPO⁶ en la India. NASSCOM es un organismo de comercio mundial con más de 1200 miembros, que incluyen compañías indias y multinacionales que tienen presencia en la India. Los miembros de NASSCOM y las compañías asociadas están por lo general en el negocio de desarrollo de software, servicios de software, productos de software, servicios de consultoría, servicios de BPO, el comercio electrónico y servicios web, servicios de ingeniería de la deslocalización y la animación y los juegos. La base de socios del NASSCOM constituye más del 95 % de los ingresos de industria en India y emplea a más de 2.24 millones de profesionales.

NASSCOM fue creado en 1988, en Mumbai para facilitar los negocios, el comercio de software y servicios y para alentar el progreso de la investigación en tecnología de software. Es una organización sin fines de lucro, registrada bajo la Ley de Sociedades de la India. En la actualidad, NASSCOM tiene su sede en Nueva Delhi, India, con oficinas regionales en las ciudades de Mumbai, Chennai, Hyderabad, Bangalore, Calcuta y Pune.

⁶ Information Technology (IT) y Business Process Outsourcing (BPO) hacen referencia a los servicios de externalización de procesos de negocios en la industria de Tecnologías de la Información en la India.

7.2.6 American National Standards Institute (ANSI)

Es una organización privada sin fines lucrativos que administra y coordina la normalización voluntaria y las actividades relacionadas a la evaluación de conformidad en los Estados Unidos. La misión del Instituto es mejorar tanto la competitividad mundial de las empresas estadounidenses, así como la calidad de vida estadounidense, promoviendo y facilitando normas voluntarias de consenso y sistemas de evaluación de conformidad, y protegiendo su integridad.

ANSI tiene como miembros 125.000 compañías y 3,5 millones de profesionales representantes del gobierno estadounidense, organizaciones, compañías, academias, intereses internacionales e individuos.

ANSI es el representante oficial de los EEUU en el International Accreditation Forum (IAF), en la International Organization for Standardization (ISO) y, a través del U.S. National Committee, en la International Electrotechnical Commission (IEC). ANSI también es el miembro representante de los EEUU en el Pacific Area Standards Congress (PASC) y en la Pan American Standards Commission (COPANT).

7.2.7 Comisión Panamericana de Normas Técnicas (COPANT)

La Comisión Panamericana de Normas Técnicas, designada por el título abreviado de COPANT, es una asociación civil sin fines de lucro. Funciona con plena autonomía y sin término de duración.

Los fines de COPANT son promover el desarrollo de la normalización técnica y actividades relacionadas en sus países miembros, con el fin de impulsar su desarrollo comercial, industrial, científico y tecnológico. COPANT busca para sus miembros el beneficio de la integración económica y comercial, del intercambio de

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

bienes y servicios y de facilitar la cooperación en las esferas intelectual, científica, económica y social.

La COPAN funciona a través de dos órganos técnicos bien diferenciados: La Comisión de Gestión Técnica (CGT), que tiene por finalidad la gestión de las actividades técnicas de COPANT; y los Comités Técnicos de Normalización que se van creando en función de los intereses de los Miembros de COPANT, por medio de los cuales se lleva a cabo el proceso de Normalización Regional.

La COPAN tiene como miembros a los principales organismos de normalización, estandarización y certificación de los diferentes países, agrupados de acuerdo a dos clasificaciones, así:

Miembros Adherentes:

- ALEMANIA - (DIN) Deutsches Institut für Normung
<http://www.din.de>
- AUSTRALIA - (SA) Standards Australia
<http://www.standards.org.au>
- Cooperación Interamericana de Acreditación (IAAC)
<http://www.iaac.org.mx>
- ESPAÑA - (AENOR) Asociación Española de Normalización y Certificación
<http://www.aenor.es>
- FRANCIA - (AFNOR) Asociación Francesa de Normalización
<http://www.afnor.fr>
- ITALIA - (UNI) Instituto Italiano de Unificación
<http://www.uni.com>
- PORTUGAL - (IPQ) Instituto Português da Qualidade
<http://www.ipq.pt>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

- SUR AFRICA - (SABS) South African Bureau of Standards
<http://www.sabs.co.za>

Miembros Activos:

- ARGENTINA - (IRAM) Instituto Argentino de Normalización y Certificación
<http://www.iram.com.ar>
- BARBADOS - (BNSI) Barbados National Standards Institution
- BOLIVIA - (IBNORCA) Instituto Boliviano de Normalización y Calidad
<http://www.ibnorca.org>
- BRASIL - (ABNT) Asociación Brasileira de Normas Técnicas
<http://www.abnt.org.br>
- CANADÁ - (SCC) Standards Council of Canada
<http://www.scc.ca>
- CHILE - (INN) Instituto Nacional de Normalización
<http://www.inn.cl>
- COLOMBIA - (ICONTEC) Instituto Colombiano de Normas Técnicas y Certificación
<http://www.icontec.org.co>
- COSTA RICA - (INTECO) Instituto de Normas Técnicas de Costa Rica
<http://www.inteco.or.cr>
- CUBA - (NC) Oficina Nacional de Normalización
<http://www.nc.cubaindustria.cu>
- ECUADOR - (INEN) Instituto Ecuatoriano de Normalización
<http://www.ecua.net.ec/inen/>
- EL SALVADOR – (CONACYT) Consejo Nacional de Ciencia y Tecnología
<http://www.conacyt.gob.sv>
- ESTADOS UNIDOS DE AMÉRICA - (ANSI) American National Standards Institute
<http://www.ansi.org>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

- GRENADA - (GDBS) Grenada Bureau of Standards
<http://www.spiceisle.com/homepages/gdbs/>
- GUATEMALA – (COGUANOR) Comisión Guatemalteca de Normas
<http://www.mineco.gob.gt>
- GUYANA - (GNBS) Guyana National Bureau of Standards
<http://www.sdn.org.gy/gnbs>
- HONDURAS - (COHCIT) Consejo Hondureño de Ciencia y Tecnología
<http://www.cohcit.gob.hn>
- JAMAICA - (BSJ) Bureau of Standards Jamaica
<http://www.bsj.org.jm>
- MÉXICO - (DGN) Dirección General de Normas/Secretaría de Economía
<http://www.economia.gob.mx/?P=85>
- NICARAGUA - (MIFIC) Ministerio de Fomento, Industria y Comercio/Dirección de Tecnología, Normalización y Metrología
<http://www.mific.gob.ni>
- PANAMÁ - (COPANIT) Comisión Panameña de Normas Industriales y Técnicas
<http://www.mici.gob.pa>
- PARAGUAY - (INTN) Instituto Nacional de Tecnología y Normalización
<http://www.intn.gov.py>
- PERÚ - (INDECOPI) Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual
<http://www.indecopi.gob.pe>
- REPÚBLICA DOMINICANA - (DIGENOR) Dirección General de Normas y Sistemas de Calidad
<http://www.digenor.gob.do>
- SANTA LUCIA – (SLBS) Saint Lucia Bureau of Standards
<http://www.slbs.org.lc>

- TRINIDAD Y TOBAGO - (TTBS) Trinidad and Tobago Bureau of Standards
<http://www.ttbs.org.tt>
- URUGUAY - (UNIT) Instituto Uruguayo de Normas Técnicas
<http://www.unit.org.uy>
- VENEZUELA - (FONDONORMA) Fondo para la Normalización y la Certificación de la Calidad
<http://www.fondonorma.org.ve>

7.2.8 International Software Benchmarking Standards Group (ISBSG)

El repositorio global e independiente de los datos y análisis para la industria de TI es una organización sin ánimo de lucro que mantiene y explota repositorios de datos históricos sobre métricas de software, para ayudar a mejorar la gestión de las tecnologías de la información.

Los miembros actuales de ISBSG representan a las asociaciones de métricas de doce países (Australia, Finlandia, China, Alemania, India, Italia, Japón, Holanda, España, Suiza, Reino Unido y Estados Unidos).

AEMES es el miembro de ISBSG correspondiente a España: Asociación Española de Métricas de Sistemas Informáticos.

ISBSG ofrece repositorios independientes de datos de la industria TI relacionados con Desarrollo de Software y Evolutivos (más de 5.100 proyectos), así como Mantenimiento de Software (más de 470 aplicaciones). Los datos de dichos repositorios se pueden utilizar para realizar estimaciones, estudios de evaluación comparativa, gestión de proyectos, planificación de infraestructura, planificación de ofertas, gestión de la externalización, cumplimiento de estándares y gestión del presupuesto.

7.2.9 Centre for Software Engineering (CSE)

CSE es el centro nacional de coordinación y servicio de apoyo a la comunidad de desarrollo de software en Irlanda, ayudándole a mejorar su calidad y productividad así como a aplicar las mejores prácticas. El Centro de Ingeniería de Software se fundó en 1990.

CSE ofrece asesoramiento y formación a medida, que las organizaciones utilizan para establecer su posicionamiento estratégico, impulsar la mejora y aumentar su capacidad de crear y explotar los productos de software de alta calidad. Los principales servicios que presta hacen referencia a la Mejora de Procesos, la Gestión de Proyectos, Arquitectura Empresarial, Ingeniería Ágil y la Innovación.

Uno de sus principales proyectos ha sido el “**Software Process Improvement in Regions of Europe (SPIRE)**”, Mejora de procesos software en las Regiones de Europa (SPIRE), financiado por la Comisión Europea dentro del programa ESPRIT/ESSI, que implica a socios de Austria, Irlanda, Italia, Suecia e Irlanda del Norte en el Reino Unido.

Su objetivo era ayudar a pequeñas unidades de desarrollo de software - SSDs, tanto empresas de software como pequeñas unidades en organizaciones de usuarios más grandes, para obtener beneficio comercial de las inversiones en mejora de procesos software (SPI) y compartir la experiencia con otros.

7.2.10 Association for Computing Machinery (ACM)

ACM es ampliamente reconocida como la principal organización de socios para los profesionales de la informática, en la entrega de recursos (principalmente de carácter técnico y documental) que permitan el avance de la informática como una ciencia y

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

una profesión, facilitando el desarrollo profesional, y promoviendo políticas de investigación que beneficien a la sociedad.

ACM alberga la más importante Biblioteca Digital y Guía de la Literatura Informática del sector, la cual sirve a sus miembros a nivel mundial y a los profesionales de la computación a través de periódicos y revistas, conferencias, talleres, foros electrónicos, libros en línea y cursos.

Su estructura organizacional está compuesta por juntas, comités y consejos regionales (Europa, India) y voluntarios. La ACM cuenta además con grupos de interés especial conocidos como SIGs (Special Interest Groups) los cuales está conformados por comunidades técnicas representantes de prácticamente todas las áreas importantes de la informática, los cuales ofrecen una gran cantidad de conferencias, publicaciones y actividades en la escala local y global, proporcionando oportunidades ilimitadas para compartir conocimientos técnicos y conocimiento de primera mano sobre las últimas tendencias de desarrollo. De los 34 grupos de interés especial con los que cuenta actualmente la ACM, se quieren destacar en el presente documento, aquellos que puedan hacer aportes significativos a la ingeniería del software, como es el caso de SIGSOFT (grupo de interés especial en Ingeniería del Software), el cual se centra en temas relacionados con todos los aspectos de desarrollo y mantenimiento de software. Sus áreas de interés especial son: requerimientos, especificación y diseño, la arquitectura de software, validación, verificación, depuración, seguridad del software, procesos de software, software de gestión, medición, interfaces de usuario, gestión de configuraciones, ingeniería de entornos de software y herramientas CASE.

SIGSOFT trata de mejorar la capacidad del ingeniero de software mediante la estimulación de la interacción entre profesionales, investigadores y educadores,

fomentando su desarrollo profesional y también mediante su representación ante entidades legales y políticas. También proporciona un foro para profesionales de la industria informática, el gobierno y académicos para examinar los principios, prácticas y nuevos resultados en la investigación de ingeniería de software.

7.2.11 European Space Agent (ESA)

La Agencia Espacial Europea es la puerta de acceso al espacio del continente europeo. Su misión consiste en configurar el desarrollo de la capacidad espacial europea y garantizar que la inversión en actividades espaciales siga dando beneficios a los ciudadanos de Europa.

La ESA está compuesta por 18 Estados Miembros: Alemania, Austria, Bélgica, Dinamarca, España, Finlandia, Francia, Grecia, Irlanda, Italia, Luxemburgo, Noruega, Países Bajos, Portugal, Reino Unido, República Checa, Suecia y Suiza. Canadá y Hungría tienen un estatus especial y participan en algunos proyectos conforme a un acuerdo de cooperación. La coordinación de los recursos económicos e intelectuales de sus miembros permite llevar a cabo programas y actividades de mayor alcance que los que podría realizar cualquier país europeo individualmente.

La misión de la ESA consiste en elaborar el programa espacial europeo y llevarlo a cabo. Los proyectos de la Agencia se diseñan con el fin de conocer más a fondo la Tierra, el entorno espacial que la rodea, el Sistema Solar y el Universo, así como para desarrollar tecnologías y servicios basados en satélites y fomentar la industria europea. La ESA también trabaja en estrecha colaboración con organizaciones espaciales no europeas, de modo que toda la humanidad pueda beneficiarse de las ventajas del espacio. (ESA, 2010)

Una sección muy importante de la ESA es aquella conocida como **Software Engineering & Standardisation**, la cual pertenece a la División de Sistemas de Software y es la responsable de los temas relacionados con la Ingeniería del Software y la Normalización de los mismos. La sección tiene experiencia en métodos de software, herramientas, arquitecturas y estándares, estando en la capacidad de proporcionar referencias detalladas de los estándares de ingeniería del software producidos por European Space Agency (ESA). Sus principales actividades están relacionadas con:

Ingeniería de software: cubren todo el ciclo de vida del software, con énfasis en las primeras fases, como co-sistema de software de ingeniería, ingeniería de requerimientos, diseño, codificación de idiomas y el medio ambiente de compilador, la automatización de la generación de código y la generación de pruebas de validación.

Estandarización de Software: implicada en la normalización de software, tanto de las normas internas para la organización (ESA) a través de la co-presidencia del BSSC (Consejo de Normalización y Control de Software), como de estándares de software externo - sobre todo con ECSS (Cooperación Europea para la Normalización Espacial) y, a veces con ISO.

Apoyo a proyectos: se encarga de apoyar a numerosos proyectos espaciales.

7.2.12 British Standards Institution (BSI)

La British Standards Institution es la entidad de normalización Británica fundada en 1901, la cual se convirtió en la primera entidad nacional de normalización del mundo y actualmente es la más grande en su tipo, produciendo una media de 2000 nuevas

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

normas cada año. Cuenta con más de 60.000 clientes registrados a nivel internacional. También es líder en certificación en el Reino Unido y Norte América.

Desde su fundación en 1901 como el comité de estándares de ingeniería, el grupo BSI presta sus servicios a las empresas de forma independiente, certificando sistemas de gestión y productos, desarrollando estándares nacionales e internacionales y proporcionando formación e información sobre estándares y comercio internacional.

BSI opera a través de sus tres unidades de negocio: BSI British Standards (estándares), BSI Management Systems (certificación de sistemas) y BSI Product Services(certificación de productos).

7.2.13 Red sobre Experimentación y Medición en Ingeniería de Software (REMIS)

REMIS fu creada bajo el auspicio de la Acción Especial CICYT TIC98-1179-E, "Coordinación de Grupos Investigadores sobre Experimentación y Medición en Ingeniería de Software" en España. Es una red formada por un grupo de docentes e investigadores de diferentes universidades españolas con interés específico en la gestión cuantitativa del software.

El objetivo principal del grupo de trabajo es el de obtener conocimiento científico en todos los aspectos relacionados con los procesos, recursos, productos y proyectos de ingeniería de software.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

7.2.14 Otros organismos de normalización a manera de información

ORGANISMOS INTERNACIONALES DE NORMALIZACIÓN		
SIGLA	NOMBRE	URL
CEI	Comisión Electrotécnica Internacional	http://www.iec.ch/
CEN	Comité Europeo de Normalización	http://www.cenorm.be/
CENELEC	Comité Europeo de Normalización Electrotécnica	http://www.cenelec.org/
CIE	Comisión Internacional de Iluminación	http://www.cie.co.at/
COPANT	Comisión Panamericana de Normas Técnicas	http://www.copant.org/
ETSI	Instituto Europeo de Normas de Telecomunicaciones	http://www.etsi.org/
ISO	Organización Internacional de Normalización	http://www.iso.org/
MERCOSUR	Comité Mercosur de Normalización	http://www.cmn.org.br/
OIML	Organización Internacional de Metrología Legal	http://www.oiml.org/
UIT	Unión Internacional de Telecomunicaciones	http://www.itu.ch/

Tabla 1: Organismos Internacionales de Normalización

ORGANISMOS DE NORMALIZACIÓN EN AMÉRICA		
SIGLA	NOMBRE	URL
IRAM	Instituto Argentino de Normalización, Argentina	http://www.iram.com.ar/
BNSI	Barbados National Standards Institution, Barbados	http://www.commerce.gov.bb/agency/bnsi/
IBNORCA	Instituto Boliviano de Normalización y Calidad, Bolivia	http://www.ibnorca.org/
ABNT	Associao Brasileira de Normas Tecnicas, Brasil	http://www.abnt.org.br/
SCC	Standards Council of Canada, Canadá	http://www.scc.ca/

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORGANISMOS DE NORMALIZACIÓN EN AMÉRICA		
SIGLA	NOMBRE	URL
INN	Instituto Nacional de Normalización, Chile	http://www3.inn.cl
ICONTEC	Instituto Colombiano de Normas Técnicas y Certificación, Colombia	http://www.icontec.org.co
INTECO	Instituto de Normas Técnicas de Costa Rica	http://www.webspawner.com/users/inteco/
NC	Oficina Nacional de Normalización.Cuba	www.nc.cubaindustria.cu/
INEN	Instituto Ecuatoriano de Normalización, Ecuador	http://www.ecua.net.ec/inen/
CONACYT	Consejo Nacional de Ciencia y Tecnología, El Salvador	www.conacyt.gob.sv/
ANSI	American National Standards Institute, Estados Unidos	http://www.ansi.org/
ASTM	American Society for Testing and Materials, Estados Unidos	http://www.astm.org/
IEEE	The Institute of Electrical and Electronics Engineers, Estados Unidos	http://www.standards.ieee.org/
GDBS	Grenada Bureau of Standards Granada	http://www.spiceisle.com/homepages/gdbs
COGUANOR	Comisión Guatemalteca de Normas, Guatemala	www.coguanor.org/
GNBS	Guyana National Bureau of Standards, Guyana	http://www.sdn.org.gy/gnbs
COHCIT	Consejo Hondureño de Ciencia y Tecnología, Honduras	www.cohcit.gob.hn/
JBS	Jamaica Bureau of Standards, Jamaica	http://www.jbs.org.jm/
DGN	Dirección general de Normas, México	http://www.secofi.gob.mx/normas/
DGICYT	Dirección General de Ciencia y Tecnología, Nicaragua	www.conicyt.gob.ni
COPANIT	Comisión Panameña de Normas Industriales y Técnicas, Panamá	http://www.mici.gob.pa/
INTN	Instituto Nacional de Tecnología y Normalización, Paraguay	www.intn.gov.py/
INDECOPI	Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual, Perú	http://www.indecopi.gob.pe/
BNQ	Bureau de Normalisation du Québec, Canadá	http://www.criq.qc.ca/bnq

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORGANISMOS DE NORMALIZACIÓN EN AMÉRICA		
SIGLA	NOMBRE	URL
DIGENOR	Dirección General de Normas y Sistemas de Calidad, República Dominicana	http://www.digenor.gob.do
TTBS	Trinidad and Tobago Bureau of Standards Trinidad y Tobago	http://www.opus.co.tt/ttbs/
UNIT	Instituto Uruguayo de Normas Técnicas Uruguay	http://www.unit.org.uy
FONDONO RMA	Fondo para la Normalización y la Certificación de la Calidad, Venezuela	http://www.fondonorma.org.ve

Tabla 2: Organismos de Normalización en América

ORGANISMOS DE NORMALIZACIÓN EN ÁFRICA		
SIGLA	NOMBRE	URL
BOBS	Botswana Bureau of standards Botswana	http://www.bobstandards.bw
CONDINORM	Côte d'Ivoire - Normalisation Costa de Marfil	http://codinorm.ci
EOS	Egyptian Organization for Standardization and Quality Control, Egipto	http://www.misrnet.idsc.gov.eg
GSB	Ghana Standads Boards Ghana	http://ghanastandards.org/
INNORPI	Institut National de la Normalisation et de la Propriété Industrielle, Tunes	http://www.innorpi.tn
KEBS	Kenya BureauStandards Kenya	http://www.kebs.org
LNCSM	Libyan National Centre for Standardization and Metrology Industrial Research, Libia	www.lncsm.org.ly
MSB	Mauritius Standard Bureau Mauricio	http://www.gov.mu/portal/sites/msb/index.htm
OCC	Office congolais de Controle Congo	http://www.occ-rdc.cd
QSAE	Quality and Standards Authority of Ethiopia, Etiopía	http://www.qsae.org

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORGANISMOS DE NORMALIZACIÓN EN ÁFRICA		
SIGLA	NOMBRE	URL
SABS	South African Bureau of Standards, Sudáfrica	https://www.sabs.co.za/
SAZ	Standards Association of Zimbabwe, Zimbawe	http://www.saz.org.zw
SBS	Seychelles Bureau of Standards Seychelles	http://www.seychelles.net/sbsorg
SONCAP	Standards Organisation of Nigeria, Nigeria	http://www.soncap.com
SSMO	Sudanese Standards and Metrology Organization, Sudán	http://www.sol-sd.com
TBS	Tanzania Bureau of Standards Tanzania	http://www.tbstz.org
UNBS	Uganda National Bureau of Standards, Uganda	http://www.unbs.go.ug

Tabla 3: Organismos de Normalización en África

ORGANISMOS DE NORMALIZACIÓN EN ASIA		
SIGLA	NOMBRE	URL
BSN	Badan Standardisasi Agency Indonesia	http://www.bsn.or.id
BSTI	Bangladesh Standard and Testing Institution Banglades	http://www.bsti.gov.bd
DSM	Department of Standards Malaysia Malasia	http://www.dsm.gov.my
GOSTR	State Committee of the Russian Federation for Standardization and Metrology, Rusia	http://www.gost.ru
ISIRI	Institute of Standards and Industrial Research of Iran, Irán	http://www.isiri.com
JISC	Japanese Industrial Standards Committee, Japón	http://www.jisc.go.jp
JISM	Jordan Institution for Standards & Metrology, Jordania	http://www.jism.gov.jo
KATS	Korean Agency for Technology and Standards, Corea del Sur	http://www.ats.go.kr
KOWSMD	Public Authority for Industry Standards and Industrial Services Affairs, Kuwait	https://www.pai.gov.kw

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORGANISMOS DE NORMALIZACIÓN EN ASIA		
SIGLA	NOMBRE	URL
LIBNOR	Lebanese Standareds Institution Líbano	http://www.libnor.org
NBSM	Nepal Bureau of Standards and Metrology Nepal	http://www.nepalstandards.gov.np
PSB	Singapore Productivity and Standards Boards Singapur	http://www.psb.gov.sg
PSI	Pakistan Standard Institution Pakistán	http://www.icap.org.pk
SARM	Department for Standardization, Metrology and Certification, Armenia	http://www.sarm.am
SASMO	Syrian Arab Organization for Standardization and Metrology, Siria	http://www.sasmo.org
SASO	Saudi Arabian Standards Organization Arabia Saudita	www.saso.org.sa
SII	Standards Institution of Israel	http://www.sii.org.il
SSUAE	Directorate of Standardization and Metrology Emiratos Arabes Unidos	http://www.uae.gov.ae
TISI	Thai Industrial Standards Institute Tailandia	http://www.tisi.go.th

Tabla 4: Organismos de Normalización en Asia

ORGANISMOS DE NORMALIZACIÓN EN EUROPA		
SIGLA	NOMBRE	URL
AENOR	Asociación Española de Normalización y Certificación España	http://www.aenor.es
AFNOR	Association Francaise de Normalisation Francia	http://www.afnor.fr
ASRO	Asociatia de Standardizare din Rumania	http://www.asro.ro
BDS	State Agency for Standardization and Metrology Bulgaria	www.bds-bg.org
BSI	British Standard Institution Reino Unido	http://www.bsi.org.uk
CSNI	Czech Standards Institute República Checa	http://www.csni.cz

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORGANISMOS DE NORMALIZACIÓN EN EUROPA		
SIGLA	NOMBRE	URL
CYS	Cyprus organization for Standards and Control of Quality Chipre	http://www.cys.mcit.gov.cy
DIN	Deutsches Institut für Normung, Alemania	http://www.din.de
DPS	General Directorate of Standardization Albania	http://www.dps.gov.al
DS	Dansk Standard Dinamarca	http://www.ds.dk
DZNM	State Office for Standardization and Metrology Croacia	http://www.dzm.hr
ELOT	Hellenic Organization for Standardization Grecia	http://www.elot.gr
EVS	National Standards Board of Estonia Estonia	http://www.evs.ee
GOSTR	State Committee of the Russian Federation for Standardization and Metrology, Rusia	http://www.gost.ru
IBN	Institut belge de Normalisation Bélgica	http://www.ibn.be
IPQ	Instituto Português de Qualidade Portugal	http://www.ipq.pt
LST	Lithuanian Standards Board Lituania	http://www.lsd.lt
LVS	Latvian Standard Letonia	http://www.lvs.lv
MOLDST	Department of Standards, Metrology and Technical Supervision, Moldavia	http://www.moldova.md
MSZT	Magyar Szabványügyi Testület Hungria	www.mszt.hu
NNI	Nederlands Normalisatie-instituut Holanda	http://www.nni.nl
NSAI	National Standards Authority of Ireland Irlanda	http://www.nsai.ie
NSF	Norges Standardiseringsforbund Noruega	http://www.standard.no
ON	Österreichisches Normungsinstitut Austria	http://www.as-institute.at
PKN	Polish Committee for Standardization Polonia	http://www.pkn.pl
SEE	Service de l'Énergie de l'Etat Département Normalisation Luxemburgo	http://www.etat.lu/SEE
SFS	Finnish Standards Association SFS Finlandia	http://www.sfs.fi

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORGANISMOS DE NORMALIZACIÓN EN EUROPA		
SIGLA	NOMBRE	URL
SIS	Standardiseringen i Sverige Suecia	http://www.sis.se
SMIS	Standards and Metrology Institute of the Republic of Slovenia, Eslovenia	http://www.usm.mzt.si
SNV	Swiss Asociation for Standardization Suiza	http://www.snv.ch
TSE	Türk Standardlari Enstitüsü Turquía	http://www.tse.org.tr
UNI	Ente Nazionale Italiano di Unificazione Italia	http://www.uni.com
UNMS	Slovak Office of Standards, Metrology and Testing Eslovenia	http://www.normoff.gov.sk
VDI	Verein Deutscher Ingenieure Alemania	http://www.vdi.de

Tabla 5: Organismos de Normalización en Europa

ORGANISMOS DE NORMALIZACIÓN EN OCEANIA		
SIGLA	NOMBRE	URL
SAI	Standards Australia International Ltd. Australia	http://www.standards.com.au
SNZ	Standards New Zealand Nueva Zelanda	http://www.standards.co.nz

Tabla 6: Organismos de Normalización en Oceanía

7.3 Organizaciones que tienen que ver con el desarrollo de software en Latinoamérica

7.3.1 CESSI (Argentina)

La Cámara de Empresas de Software y Servicios informáticos (CESSI), es la entidad que nuclea y representa a las empresas dedicadas al desarrollo, producción,

comercialización e implementación del software y todas las variantes de servicios en todo el ámbito de la República Argentina.

Con más de 300 empresas o entidades asociadas actualmente representa el 80% de los ingresos del sector y más del 85% de los empleos. Como tal, CESSI, hoy en día, es uno de los referentes centrales de la Industria de Tecnología Argentina entre los gobiernos nacional y extranjeros, el sector privado, la academia y los mercados Internacionales.

CESSI cuenta con doce comisiones de trabajo, dentro de las cuales se destaca la Comisión de Calidad, que tiene como misión promover activamente la implantación de estándares de calidad en la producción de software, orientados a establecer patrones de desempeño mínimo de modo que fortalezcan la capacidad de producción de las empresas, mejoren los procesos de desarrollo y potencien la inserción internacional de empresas locales.

7.3.2 SOFTEX (Brasil)

Sociedad Brasileña para la Promoción de la Exportación de Software – SOFTEX creada a finales de 1996, como una organización no gubernamental cuyo objetivo social es el de realizar, promover, fomentar y apoyar las actividades de innovación y desarrollo científico y tecnológico de generación y transferencia de tecnologías y principalmente de promoción del capital humano, a través de la educación, la cultura y la formación adecuada, de naturaleza técnica y de marketing de software y sus aplicaciones, con énfasis en el mercado externo, la orientación del desarrollo socioeconómico de Brasil, a través de la integración del país con la economía mundial.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

A finales del 2004 SOFTEX cambió su nombre a: Asociación para la Promoción de la Excelencia del Software Brasileño - SOFTEX, organización de la Sociedad Civil de Interés Público con sede en Campinas.

El Sistema SOFTEX posee acciones y metas en concordancia con las siguientes directrices estratégicas:

- 1. Difundir y contribuir a la aplicación de mejores prácticas en la Industria Brasileña de Software y Servicios:** actividades de promoción de las mejores prácticas – la calidad, las asociaciones, la gestión, entre otras – en las empresas brasileñas de software y servicios adhiriéndose a las normas internacionales.
- 2. Apoyar la creación y desarrollo de oportunidades de negocios para la Industria Brasileña de Software y Servicios:** Actividades que amplíen y consoliden la participación de las empresas brasileñas de software y servicios en los mercados nacional e internacional.
- 3. Apoyar la capacitación de recursos humanos para la industria brasileña de software y servicios:** actividades de capacitación gerencial, mercadológica y técnica de los profesionales en las (y para las) empresas brasileñas de software y servicios.
- 4. Apoyar el apalancamiento de recursos financieros para la industria brasileña de software y servicios:** las actividades de articulación y apoyo a las empresas brasileñas de software y servicios para recaudar fondos dirigidos al desarrollo de tecnologías y negocios.

5. **Producir y difundir información de calidad calificada para la industria brasileña de software y servicios:** actividades de obtención, generación, análisis y suministro de información sobre la Industria de Software y Servicios en Brasil y en el mundo.

6. **Apoyo la formulación de políticas públicas de interés de la Industria Brasileña de Software y Servicios:** acciones que promuevan la participación del Sistema SOFTEX en la formulación y ejecución de políticas para el sector.

7. **Apoyar el espíritu empresarial en la Industria Brasileña de Software y Servicios:** actividades relacionadas con la creación de empresas brasileñas de software y servicios.

SOFTEX creó en el año 2004 el Programa MPS.BR (Mejoría del Proceso del Software Brasileño), cuyo objetivo principal es desarrollar y difundir un modelo de proceso de software destinado a mejorar los procesos de software de las organizaciones de Brasil en un plazo razonable dentro de costos razonables (Santos, Chaves Weber, & Cavalcanti da Rocha, 2009).

7.3.3 GECHS (Chile)

Sociedad Chilena de Software y Servicios A.G. Fundada en el año 2002, con la misión de promover el desarrollo de la industria de software y servicios relacionados, buscando que sus empresas asociadas logren posicionar sus productos y servicios en el ámbito nacional e internacional. Gechs reúne a cerca de 70 empresas nacionales en las áreas de Productos de Software, Ingeniería de Software, Outsourcing de Personal TI, Consultoría TI y Consultoría en Procesos.

Gechs fue constituida para cumplir los siguientes objetivos: estudiar, investigar, desarrollar, fomentar y promocionar la industria de la creación y diseño de programas computacionales y los servicios anexos derivados de la misma industria. Esta organización en conjunto con Escuela de Ingeniería de la Universidad de Viña del Mar y Microsoft Chile, ha realizado en los últimos años una serie de estudios diagnósticos de la industria de software y servicios en Chile, que le han permitido adquirir una mejor visión del grado de avance, madurez y complejidad del sector, desde el punto de vista de los productos y sus mercados, los procesos, las personas, la investigación y la economía.

7.3.4 CAMTIC (Costa Rica)

La Cámara Costarricense de Tecnologías de Información y Comunicación es una asociación empresarial, privada y sin fines de lucro establecida en 1998. Su razón de ser fue formar un bloque estratégico que permitiera fortalecer y apoyar al sector de tecnologías de información y comunicación (TIC). CAMTIC inició sus actividades solo representando al sector software, hoy agrupa a más del 90% de todas las empresas nacionales de software.

Ante la reciente ampliación en pro del restante sector de TIC, hoy comienzan a incorporarse empresas de los otros segmentos TIC. La organización es la única en su categoría en Centroamérica.

Actualmente ocupa la Vicepresidencia Ejecutiva de la Asociación Latinoamericana de Entidades de Tecnologías de Información (ALETI), de la cual es también fundadora. Además es miembro activa de la Federación Internacional de Asociaciones de Multimedia (FIAM) y de la Alianza Mundial de Tecnologías de Información y Servicios (WITSA).

7.3.5 AESOFT (Ecuador)

La Asociación Ecuatoriana de Software, **AESOFT**, es una organización gremial privada sin fines de lucro creada en mayo de 1995 en Quito, Ecuador. Busca aglutinar a las empresas de la industria de tecnologías de información y comunicaciones propendiendo a alcanzar el desarrollo tecnológico del país. Agrupa a empresas productoras, distribuidoras y desarrolladoras de software así como compañías dedicadas a la prestación de servicios informáticos relacionados con el software y la tecnología. (AESOFT, 2009)

Tiene como fines impulsar, fomentar e incentivar, el desarrollo y perfeccionamiento tecnológico en el diseño, desarrollo, producción, personalización, comercialización y explotación de los programas de computación (software) y sistemas informáticos, de sistemas de información y comunicación relacionados, buscando alcanzar por estos medios el crecimiento de la industria de tecnologías de información.

7.3.6 SOFEX (Guatemala)

La Comisión de Software de Exportación SOFEX, es una organización creada con la misión de promover y apoyar a la industria de Tecnologías de Información de Guatemala, para hacerla globalmente más competitiva, reconocida exitosamente y sustentable por medio de la participación, integridad y productividad de sus socios. Está conformada por empresas dedicadas al desarrollo de software, las cuales ofrecen productos y servicios altamente calificados para atender el mercado local e internacional de una manera competitiva e innovadora. Dentro de sus objetivos, se destaca el apoyo a los procesos de mejora continua aumentando la calidad de los productos de Software guatemalteco.

7.3.7 APESOFT (Perú)

Asociación Peruana de Productores de Software (APESOFT), es una entidad privada sin fines de lucro, fundada en el año 2000 con el objetivo de promover la industria nacional del software, mejorar la competitividad de sus afiliados y fomentar las exportaciones de programas informáticos peruanos.

APESOFT está constituida por las principales empresas desarrolladoras de software, algunas de las cuales han logrado obtener certificaciones de calidad bajo estándares internacionales, como ISO9000, IT MARK⁷, siendo el interés de la asociación el que todos sus afiliados cuenten con algún tipo de acreditación en calidad a fin de mejorar la competitividad del software peruano. (APESOFT, 2009)

APESOFT con la cooperación de la Corporación Andina de Fomento – CAF⁸ tiene como propósito llevar a cabo el proyecto denominado “**DESCENTRALIZACION DE LA CALIDAD PARA LA COMPETITIVIDAD DEL SOFTWARE**”, cuyo objetivo es descentralizar la cultura de la calidad llegando a empresas de software localizadas en las principales regiones del país, a través de la creación, difusión y adopción del modelo adaptado de CMMI. Sus beneficiarios directos serán el conjunto de empresas productoras de software localizadas en las principales provincias del país, estimadas

⁷ http://www.fedesoft.org/downloads/Sinertic/RESUMEN_MODELO_IT_MARK.pdf. IT Mark es el primer modelo de calidad internacional diseñado específicamente para las pequeñas y medianas empresas. Es un modelo escalable y muy adecuado para las Pequeñas y Micro Empresas del sector TIC.

⁸ <http://www.caf.com/view/index.asp?pageMS=34132&ms=17>. La CAF es una institución financiera multilateral, que presta múltiples servicios bancarios a clientes de los sectores público y privado de sus países accionistas, mediante la eficiente movilización de recursos financieros desde los mercados internacionales hacia América Latina. La Institución está comprometida con el desarrollo sostenible y la integración regional, pilares de su misión.

en un universo de 20 empresas, de las cuales 8 empresas, en su calidad de empresas piloto, recibirán entrenamiento y asistencia técnica en la implantación del modelo CMMI Nivel 2 adaptado a la realidad peruana. La actividad de capacitación en calidad de software beneficiará también a un grupo de 75 profesionales en tecnología de información vinculados a las universidades regionales. Finalmente, el programa de consultoría en el nuevo modelo CMMI adaptado, que será desarrollado con este proyecto, servirá a la institución APESOFT como modelo a ser replicado en otras empresas pymes del sector.

7.3.8 AMITI (México)

La Asociación Mexicana de la Industria de Tecnologías de la Información (AMITI) es una organización privada creada para posicionar la Tecnología de la Información como el ingrediente clave para la competitividad de México, promoviendo el crecimiento de la industria mediante la búsqueda de un marco normativo y jurídico que facilite el desarrollo empresarial. (AMITI, 2010)

AMITI fue fundada inicialmente en 1985 como la Asociación de la Industria software, en 1997 se convirtió en AMITI, que incorporó, además de software, otros sectores tales como los fabricantes de hardware, integradores, consultores, canales y proveedores de servicios.

Su misión es tomar el liderazgo en la industria de TI, para colaborar con el Gobierno, la academia y los usuarios para promover el uso de la estructura de información de los países para facilitar la competitividad de los mismos, así como el crecimiento de la industria y sus participantes.

AMITI representa a más de 250 empresas de TI en México, entre las que se encuentran los fabricantes de hardware y proveedores, desarrolladores de software y

vendedores, proveedores de servicios, integradores y canales. Dentro de los agremiados de la AMITI, se pueden destacar algunos de los miembros de honor del gobierno, universidades y embajadas de países extranjeros.

7.3.9 CAVEDATOS (Venezuela)

La Cámara Venezolana de Empresas de Tecnologías de la Información (CAVEDATOS), es una Asociación Civil de carácter privado, autónoma y sin fines de lucro, de interés colectivo y apolítica. Actualmente es el representante nacional del sector privado de industria y comercio relacionado con fabricación e integración de software, hardware y redes, incluyendo consultoría en tecnologías de información, Internet y otras áreas complementarias de las comunicaciones y la informática.

CAVEDATOS es la asociación que representa, fomenta, desarrolla, defiende y protege al sector privado de las tecnologías de la información en cualquiera de sus áreas. (CAVEDATOS, 2010)

7.3.10 CUTI (Uruguay)

La Cámara Uruguaya de Tecnologías de la Información (CUTI) es la organización que representa a la industria uruguaya de tecnologías de la información (TI), reconocida y legitimada en el país y el exterior por las empresas, la sociedad y el Estado. Está integrada por casi 300 firmas y organizaciones que operan exitosamente en 52 mercados y brindan soluciones para los cinco continentes.

CUTI fue fundada en el año 1989 por las más pujantes empresas productoras de software locales, con el firme propósito de desarrollar y promover esta industria sobre bases sólidas.

La organización es centro de referencia y partner activo de instituciones relacionadas

con las TI, como incubadoras de empresas, centros académico-industriales, universidades, parques tecnológicos, gremiales y asociaciones.

Tiene como misión impulsar el desarrollo sostenible del sector de tecnologías de la información y las comunicaciones, dinamizando los mercados, facilitando el crecimiento y globalización de sus miembros y poniendo énfasis en el desarrollo de las personas y la responsabilidad social. (CUTI, 2010)

7.3.11 CTIP (Paraguay)

La Cámara de Tecnologías de la Información del Paraguay (CTIP) es una asociación civil sin fines de lucro, de carácter apolítico, constituida por personas físicas y jurídicas interesadas en el área de la Tecnología de la Información.

CTIP tiene como misión promover el desarrollo de Tecnologías de la Información con calidad para exportación basado en el intercambio de información, experiencias y conocimientos entre las empresas nacionales e internacionales. (CTIP, 2010)

7.3.12 CBTI (Bolivia)

La Cámara Boliviana de Tecnologías de la Información (CBTI) es el órgano encargado del fomento de la industria y el mercado de las Tecnologías de la Información (TI) como estrategia para aumentar la competitividad del país, planteando un escenario adecuado al potenciamiento y el desarrollo del sector alineado a los requerimientos de las empresas afiliadas a la CBTI, las necesidades y demanda del país que permitan competitividad global. Su principal objetivo es impulsar a la industria de software y extender el mercado de tecnologías de información tanto a nivel nacional como internacional, articulando los esfuerzos de diversas instancias e instituciones, nacionales como internacionales, para el

desarrollo planificado y competitivo del sector, posicionando a Bolivia como un referente internacional de la industria de Software en los próximos cinco años.

La misión de CBTI es integrar y alinear las acciones de los diversos actores del sector de Tecnología de la Información y Comunicación para impulsar su desarrollo sostenible y competitivo a nivel internacional y el progreso socioeconómico, canalizando de manera eficiente la cooperación internacional para el desarrollo de la industria de las TICS en el país con responsabilidad social. (CBTI, 2010)

7.3.13 CONACYT (El Salvador)

El Consejo Nacional de Ciencia y Tecnología es una Institución de Derecho Público sin fines de lucro, de carácter autónomo descentralizado, que será la autoridad superior en materia de política científica y tecnológica. Tiene como misión formular, dirigir, coordinar y difundir continuamente las políticas nacionales de la ciencia y de la tecnología; orientadas al desarrollo económico, social y ambiental del país.

CONACYT tiene como principal objetivo formular y dirigir la política nacional en materia de desarrollo científico y tecnológico, orientada al desarrollo económico y social del país. (CONACYT, 2010)

7.4 Organizaciones que tienen que ver con el desarrollo de software en Colombia

7.4.1 FEDESOFTE (Colombia)

La Federación Colombiana de la Industria del Software (FEDESOFTE), es la entidad gremial con mayor representatividad del sector de las Tecnologías de la Información,

que agremia a la Industria de Software y Tecnologías Informáticas Relacionadas, con el objetivo de representar sus intereses ante entidades públicas y privadas, a nivel nacional e internacional. (FEDESOFTE, 2009)

En la actualidad FEDESOFTE cuenta con más de 180 afiliados en toda Colombia, sostiene convenios con Entidades del Gobierno, Universidades y otras Asociaciones a fin de continuar con el avance exponencial del sector promoviendo la Campaña Antipiratería, los Programas de Educación Continuada, El Proceso de Certificación de la Gestión de Calidad, y el apoyo a todo lo relacionado con el progreso y bienestar de sus afiliados.

7.5 Conceptualización

Con el fin de presentar una visión más clara de los conceptos relacionados con el establecimiento y mejoramiento de la calidad del software en el ámbito nacional e internacional, se considera importante definir de acuerdo a algunos autores reconocidos y organizaciones que tratan la disciplina, aquellas prácticas de ingeniería que posibilitan el cumplimiento de los lineamientos establecidos para garantizar que un producto software satisfaga las necesidades de los usuarios y conserve una estructura óptima y eficaz para alcanzar satisfactoriamente dicho propósito.

7.5.1 Norma

De acuerdo a Saravia, una norma establece un marco de referencia común para los procesos del ciclo de vida del software con una terminología bien definida a la que puede hacer referencia la industria del software. Contiene procesos, actividades y tareas para aplicar durante la adquisición de un sistema que contiene software, un

producto software puro o un servicio software y durante el suministro, desarrollo, operación y mantenimiento de productos software.

Una norma incluye también un proceso que se puede emplear para definir, controlar y mejorar los procesos del ciclo de vida del software (Saravia Aramayo, 2007, pág. 4).

“Una norma es un documento que contiene una serie de reglas, especificaciones técnicas, y características optimizadas, que se ha elaborado y aprobado por consenso a través de un organismo reconocido internacionalmente” (El Mercado CE).

7.5.2 Estándar

Documento, establecido por consenso y aprobado por un organismo reconocido, que proporciona para un uso común y repetido una serie de reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo en un contexto dado. (ISO/IEC, 2010)

7.5.2.1 Estándar Internacional.

“Estándar que ha sido adoptado por una organización internacional de estandarización/normalización y puesta a disposición del público” (ISO/IEC, 2010).

7.5.3 Regulación

“Documento que proporciona las leyes que son adoptadas por una autoridad” (ISO/IEC, 2010).

7.5.3.1 Regulación Técnica.

“Regulación que establece los requisitos técnicos, ya sea directamente o por referencia a la incorporación del contenido de un estándar, especificación técnica o código de buenas prácticas” (ISO/IEC, 2010).

7.5.4 Modelo de Procesos de Software

Según Sommerville, un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería de software.

La mayor parte de los modelos de procesos del software se basan en uno de los tres modelos generales o paradigmas de desarrollo de software: El Enfoque en Cascada, Desarrollo Iterativo y la Ingeniería del Software Basada en componentes (CBSE). (Sommerville, 2005, págs. 60,61)

Para Weitzenfeld, *“El modelo se puede definir como un grupo de estrategias, actividades, métodos y tareas, que se organizan para lograr un conjunto de metas y objetivos”.* (Weitzenfeld, 2002, cap. 3, pág. 1)

Un modelo de procesos de software se selecciona de acuerdo a la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren.

7.5.5 Metodología

“Una metodología es un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software” Piattini Velthuis (Citado en Cataldi, 2000).

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

De acuerdo a Weitzenfeld, *“una metodología se refiere al estudio de los métodos, existiendo un gran número de metodologías para el desarrollo de software. En general, distintas metodologías llevan a cabo las actividades del desarrollo de software de diferente manera”* (Weitzenfeld, 2002, cap. 3, pág. 2).

Según Rumbaugh, una metodología de ingeniería software es un proceso para la producción organizada del software, empleando una colección de técnicas predefinidas y convencionales en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso... Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo. Rumbaugh (Citado en Fernández Lanvin, 2004, pág. 32)

Algunos autores emplean el término *proceso de desarrollo de software* para denominar, sino exactamente, prácticamente el mismo concepto. Para (S. Pressman, 2002, cap. 4, pág. 13), se trata de *“un marco de trabajo de las tareas que se requieren para construir software de alta calidad”*. Según (Sommerville, 2005, pág. 7), se trata del *“conjunto de actividades y resultados asociados que producen un producto de software”*. La IEEE aporta una definición más global: *“aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería de software”* (Fernández Lanvin, 2004, pág. 32).

Para Cataldi, una metodología es un conjunto de componentes que especifican:

- Cómo se debe dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué salidas se producen y cuándo se deben producir.
- Qué restricciones se aplican.

- Qué herramientas se van a utilizar.
- Cómo se gestiona y controla un proyecto.

Normalmente consistirá en fases o etapas descompuestas en subfases, módulos, etapas, pasos, etc. Esta descomposición ayuda a los desarrolladores en la elección de las técnicas a utilizar en cada estado del proyecto, facilitando la planificación, gestión, control y evaluación de los proyectos. (Cataldi, 2000, pág. 26)

Es importante tener en cuenta, que una metodología puede seguir uno o varios modelos de ciclos de vida, los cuales indican qué es lo que hay que obtener a lo largo del desarrollo del proyecto más no especifican cómo. Es por este motivo, que la metodología se encarga de dar las pautas necesarias para que los desarrolladores tengan claro cómo alcanzar los objetivos propuestos.

7.5.6 Guía

“Una guía es un documento que contiene las directrices sobre cómo realizar los procesos” (García, Garzás, & Piattini, pág. 3).

8. PRESENTACIÓN Y ANÁLISIS INDIVIDUALIZADO DE RESULTADOS

Tomando como punto de partida las diferentes prácticas de ingeniería del software utilizadas en el ámbito internacional y con el fin de contribuir en la creación de un estándar para Colombia, que permita desarrollar, evaluar y mantener software de calidad altamente competitivo, se presentan a continuación los aspectos más relevantes de cada una de las propuestas consideradas en la investigación, resaltando aquellos que podrían llegar a ser importantes en la elaboración del mismo.

8.1 Estándares y normas

8.1.1 Características de calidad

Según Weitzenfeld, la *calidad de software* significa diferentes cosas para distintos grupos. Para la IEEE la calidad de software es el grado en que un sistema, componente o proceso cumple con los requerimientos especificados y con las necesidades o expectativas del cliente o usuario [American National Standard, 1984].

En la definición de la norma ISO 9000, la calidad de software es el grado (pobre, bueno o excelente) en que un conjunto de características inherentes del software cumplen con los requisitos. (Weitzenfeld, 2002, cap. 3, pág. 6)

Por su parte, Sommerville proporciona la siguiente definición: “*Calidad significa que un producto debe encontrar su especificación*” (Sommerville, 2005)

La norma ISO/IEC 9126 define el concepto de calidad de la siguiente manera: “*El conjunto total de características de una entidad (producto, proceso o servicio) que le*

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

confieren la capacidad de satisfacer las necesidades establecidas y las necesidades implícitas” (Granollers, 2004, pág. 42).

ISO/IEC 9126 -1: Ingeniería de Software - Calidad del producto - Modelo de Calidad: 2001			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input checked="" type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	Se incluye en la investigación porque para la Ingeniería del Software, la calidad en los productos software es un aspecto muy importante. Además, porque se considera trascendental que en el nuevo estándar se tengan en cuenta modelos para evaluar la calidad del software.		
PROPÓSITO	Evaluar la calidad del producto software.		
CONTENIDO	<p>Establece un estándar internacional para la evaluación de la calidad de productos de software en el cual se indican las características de calidad que el mismo debe tener.</p> <p>Define la calidad interna y externa de una aplicación basada en los atributos de funcionalidad, fiabilidad, usabilidad, eficiencia, mantenimiento y portabilidad y la calidad en el uso basada en las características de efectividad, productividad seguridad y satisfacción.</p> <p>Define además un marco de trabajo para el modelo de calidad que explica la relación entre diferentes aproximaciones de calidad en el ciclo de vida del desarrollo software.</p> <p>Provee un entorno para que las organizaciones definan un modelo de calidad para el producto software.</p>		
VENTAJAS	<ul style="list-style-type: none"> Este estándar está pensado para los desarrolladores, adquirentes, personal que asegure la calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC 9126 -1: Ingeniería de Software - Calidad del producto - Modelo de Calidad: 2001	
VENTAJAS	<ul style="list-style-type: none"> ▪ En conjunto con las otras partes del estándar, puede servir para validar la completitud de una definición de requisitos, identificar requisitos de calidad de software, objetivos de diseño y prueba, criterios de aseguramiento de la calidad, etc. ▪ La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del código cuando se ejecuta), por medio de las características que propone el modelo de calidad.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Al ser un estándar internacional que proporciona un marco para definir un modelo de calidad para un producto de software, se hace necesario que las organizaciones que lo aplican establezcan su propio modelo de acuerdo a las características del entorno y el tipo de software a realizar. ▪ Al ser un estándar aplicado a nivel mundial, es factible que se interprete de manera errónea y por ende, se aplique inadecuadamente en alguna organización.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque es necesario que la nueva propuesta permita evaluar la calidad del producto software teniendo como base lineamientos internacionales, que lo hagan altamente competitivo respecto a los desarrollos realizados en otros países. Además, porque se considera que la calidad de cualquier proceso del ciclo de vida del software (estándar ISO 12207) influye en la calidad del producto software que, a su vez, contribuye a mejorar la calidad en el uso del producto.
Tabla 7: Análisis de la Norma ISO/IEC 9126-1 (2001)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC TR 9126 - 2: Ingeniería de Software - Calidad de producto – Métricas Externas							
ORGANIZACIÓN	International Organization for Standardization (ISO)						
TIPO	<table border="1"> <tr> <td>Estándar <input type="checkbox"/></td> <td>Norma <input checked="" type="checkbox"/></td> <td>Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Se incluye en la investigación porque se considera importante que el nuevo estándar tenga en cuenta las métricas externas para medir las seis características de calidad definidas en la norma ISO/IEC 9126-1.						
PROPÓSITO	Proporcionar métricas externas para la medición de los atributos de las seis características de calidad externa definidas en la norma ISO/IEC 9126-1.						
CONTENIDO	<p>La norma contiene:</p> <ul style="list-style-type: none"> ▪ Una explicación de cómo aplicar las métricas de calidad de software. ▪ Un conjunto básico de indicadores para cada subcaracterística. ▪ Un ejemplo de cómo aplicar las métricas de software durante el ciclo de vida del producto. 						
VENTAJAS	<ul style="list-style-type: none"> ▪ Define parámetros externos (métricas externas para medir el comportamiento del sistema basado en computadoras que incluye el software). 						
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Al ser una norma aplicada a nivel mundial, es factible que se interprete de manera errónea y por ende, se aplique inadecuadamente en alguna organización. 						
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
¿POR QUÉ?	Porque si la nueva propuesta marco para Colombia va a tener en cuenta el Estándar ISO/IEC 9126-1, es necesario que también se incluya la Norma ISO/IEC TR 9126-2 ya que ésta es un complemento para la aplicación y entendimiento de la primera en mención.						

Tabla 8: Análisis de la Norma ISO/IEC TR 9126-2 (2003)

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC TR 9126 - 3: Ingeniería de Software - Calidad de producto – Métricas Internas							
ORGANIZACIÓN	International Organization for Standardization (ISO)						
TIPO	<table border="1"> <tr> <td>Estándar <input type="checkbox"/></td> <td>Norma <input checked="" type="checkbox"/></td> <td>Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Se incluye en la investigación porque se considera importante que el nuevo estándar tenga en cuenta las métricas internas para medir las seis características de calidad definidas en la norma ISO/IEC 9126-1.						
PROPÓSITO	Proporcionar indicadores internos de medición de los atributos de las seis características de calidad externa definidas en la norma ISO/IEC 9126-1.						
CONTENIDO	<p>La norma contiene:</p> <ul style="list-style-type: none"> ▪ Una explicación de cómo aplicar las métricas de calidad de software. ▪ Un conjunto básico de indicadores para cada subcaracterística. ▪ Un ejemplo de cómo aplicar las métricas de software durante el ciclo de vida del producto. 						
VENTAJAS	<ul style="list-style-type: none"> ▪ Está dirigida a compradores, evaluadores, desarrolladores, mantenedores, proveedores, usuarios y gerentes de calidad. ▪ Las métricas internas aplican a un producto de software no ejecutable. <ul style="list-style-type: none"> • Aplican durante las etapas de su desarrollo. • Permiten medir la calidad de los entregables intermedios. • Permiten predecir la calidad del producto final. • Permiten al usuario iniciar acciones correctivas temprano en el ciclo de desarrollo. 						
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Al ser una norma aplicada a nivel mundial, es factible que se interprete de manera errónea y por ende, se aplique inadecuadamente en alguna organización. ▪ Carencia de un glosario inglés-Español. 						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC TR 9126 - 3: Ingeniería de Software - Calidad de producto – Métricas Internas	
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque si la nueva propuesta marco para Colombia va a tener en cuenta el Estándar ISO/IEC 9126-1, es necesario que también se incluya la Norma ISO/IEC TR 9126-3 ya que ésta es un complemento para la aplicación y entendimiento de la primera en mención.
Tabla 9: Análisis de la Norma ISO/IEC TR 9126-3 (2003)	

ISO/IEC TR 9126 - 4: Ingeniería de Software - Calidad de producto – Calidad en Métricas de Uso							
ORGANIZACIÓN	International Organization for Standardization (ISO)						
TIPO	<table border="1"> <tr> <td>Estándar <input type="checkbox"/></td> <td>Norma <input checked="" type="checkbox"/></td> <td>Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Se incluye en la investigación porque se considera importante que el nuevo estándar tenga en cuenta la calidad de las métricas de uso para medir los atributos definidos en la norma ISO/IEC 9126-1.						
PROPÓSITO	Proporcionar calidad en las métricas para medir el uso de los atributos definidos en la norma ISO/IEC 9126-1.						
CONTENIDO	<p>La norma contiene:</p> <ul style="list-style-type: none"> ▪ Una explicación de cómo aplicar las métricas de calidad de software. ▪ Un conjunto básico de indicadores para cada subcaracterística. ▪ Un ejemplo de cómo aplicar las métricas de software durante el ciclo de vida del producto. 						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC TR 9126 - 4: Ingeniería de Software - Calidad de producto – Calidad en Métricas de Uso	
VENTAJAS	<ul style="list-style-type: none"> Los desarrolladores, evaluadores, directores de calidad y los compradores pueden seleccionar parámetros de la norma ISO/IEC TR 9126-4 para la definición de necesidades, evaluación de productos de software, la medición de aspectos de calidad y otros fines.
DESVENTAJAS	<ul style="list-style-type: none"> Al ser una norma aplicada a nivel mundial, es factible que se interprete de manera errónea y por ende, se aplique inadecuadamente en alguna organización.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p style="text-align: center;">Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>
¿POR QUÉ?	<p>Porque si la nueva propuesta marco para Colombia va a tener en cuenta el Estándar ISO/IEC 9126-1, es necesario que también se incluya la Norma ISO/IEC TR 9126-4 ya que ésta es un complemento para la aplicación y entendimiento de la primera en mención.</p>
Tabla 10: Análisis de la Norma ISO/IEC TR 9126-4 (2004)	

ISO/IEC 14598: Evaluación del Producto Software			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input checked="" type="checkbox"/> Modelo <input type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input type="checkbox"/>	Guía <input type="checkbox"/>
ANALIZADA EN LA INVESTIGACIÓN	<p style="text-align: center;">Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>		
JUSTIFICACIÓN	Es una estándar que procura que tanto el desarrollo como la adquisición del software, cumplan las expectativas y necesidades del usuario.		
PROPÓSITO	Proveer un marco de trabajo para evaluar la calidad de todos los tipos de productos de software y establecer los requisitos para métodos de evaluación y medición de los productos software.		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC 14598: Evaluación del Producto Software	
CONTENIDO	<p>Ofrece una visión general, explica la relación general entre su serie y el modelo de calidad de la ISO/IEC 9126, define los términos técnicos usados, contiene requisitos generales para la especificación y evaluación de la calidad del software, y clarifica los conceptos generales. Además, provee un marco de trabajo para evaluar la calidad de todos los tipos de productos de software.</p> <p>Consta de seis partes:</p> <p>ISO/IEC 14598 – Parte 1: Visión General ISO/IEC 14598 – Parte 2: Planificación y Gestión ISO/IEC 14598 – Parte 3: El Proceso para Desarrolladores ISO/IEC 14598 – Parte 4: El Proceso para Compradores ISO/IEC 14598 – Parte 5: El Proceso para Evaluadores ISO/IEC 14598 – Parte 6: Documentación de los Módulos de Evaluación</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Las seis partes que la componen describen los requisitos del proceso de evaluación en tres situaciones diferentes: Requisitos para desarrolladores (parte 3), requisitos para compradores (parte 4) y requisitos para evaluadores (parte 5). ▪ Permite establecer el propósito de la evaluación (productos intermedios, producto final). ▪ Parte de éste estándar puede usarse conjuntamente con la ISO/IEC 9126-1
DESVENTAJAS	Dentro de su proceso de evaluación no incluye las métricas externas, ni las métricas de calidad en uso del producto software.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque considera aspectos diferentes a los que tiene en cuenta la norma ISO/IEC 9126 que también son importantes a la hora de evaluar la calidad del producto software, tales como: El proceso de evaluación y el apoyo a la evaluación de los recursos y el entorno.
Tabla 11: Análisis del Estándar ISO/IEC 14598 (1998-2001)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO 25000: Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE)			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input checked="" type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	Constituye el compendio de estándares y normas internacionales más recientes para evaluar la calidad del software.		
PROPÓSITO	Proporcionar una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE), las cuales constituyen una serie de normas basadas en la ISO 9126 (Calidad del producto software) y en la ISO 14598 (Evaluación del producto software), y su objetivo principal es guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad.		
CONTENIDO	<p>La familia de normas SQuaRE está dividida en cinco partes denominadas "divisiones" por el estándar. Estas cinco divisiones son las siguientes:</p> <p>ISO 2500n - Gestión de calidad: Define todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.</p> <p>ISO 2501n - Modelo de calidad: Presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.</p> <p>ISO 2502n - Mediciones de calidad: Incluyen un modelo de referencia de calidad del producto software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.</p> <p>ISO 2503n - Requisitos de calidad: Ayudan a especificar los requisitos de calidad para ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado ó como entrada para un proceso de evaluación.</p>		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO 25000: Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE)	
CONTENIDO	ISO 2504n - Evaluación de la calidad: Proporciona requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.
VENTAJAS	<ul style="list-style-type: none"> ▪ Contiene una explicación del proceso de transición entre la antigua norma ISO/IEC 9126, la serie 14598 y SQuaRE. ▪ Presenta información sobre cómo utilizar la norma ISO/IEC 9126 y la serie 14598 en su forma anterior. ▪ Está formada por divisiones, lo que le permite encargarse de manera independiente de diferentes aspectos. ▪ Presenta estándares de extensión SQuaRE (ISO/IEC 25050–25099), los cuales incluyen requisitos para la calidad de productos de software y para el formato común de la industria para informes de usabilidad.
DESVENTAJAS	No establece los niveles de calidad deseables para cada proyecto, si bien se recomienda que los requisitos de calidad deberán ser proporcionales a las necesidades de la aplicación y lo crítico que sea el correcto funcionamiento del sistema implementado.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	<p>Porque trata de propiciar la unificación de normas para la especificación de requisitos y la evaluación de la calidad, ya que ésta división dio lugar a inconsistencias entre ambos estándares debido a la utilización de ciclos de vida separados para la especificación de requisitos y para la evaluación del producto.</p> <p>Por lo anterior, se considera importante que el conjunto de estándares SQuaRE sea tenido en cuenta para la elaboración de la nueva propuesta, ya que el mismo está orientado a reunificar los procesos mencionados utilizando como base un proceso de medida de la calidad, reemplazando tanto a ISO/IEC 9126 como a ISO 14598.</p>
Tabla 12: Análisis de la norma ISO 25000 (2005)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/TR 9241-100: Introducción a las Normas relacionadas con la Ergonomía del Software			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	Es importante que los desarrolladores de software que utilizan las normas relacionadas con la ergonomía del mismo, puedan identificar las que son verdaderamente relevantes para construir un producto de calidad que satisfaga plenamente los requerimientos hechos por el usuario.		
PROPÓSITO	Permitir a los usuarios de las normas relacionadas con la ergonomía de software identificar aquellas que son relevantes para el desarrollo del mismo, obtener una visión general sobre su contenido y entender su papel en la especificación de los requisitos de usuario, así como en el diseño y evaluación de interfaces y comprender la relación entre las distintas normas.		
CONTENIDO	La norma contiene una introducción a las normas relacionadas con la Ergonomía del Software, una descripción general de su contenido y observaciones sobre la relación entre dichas normas.		
VENTAJAS	Son aplicables a todos los componentes de software de un sistema interactivo afectando la usabilidad, incluyendo software de aplicación, sistemas operativos, software embebido, herramientas de desarrollo de software y tecnologías de asistencia.		
DESVENTAJAS	Muestra al desarrollador las normas relacionadas con la ergonomía del software, mas no indica cómo implementarlas.		
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/TR 9241-100: Introducción a las Normas relacionadas con la Ergonomía del Software	
¿POR QUÉ?	Porque es importante que en la propuesta marco para Colombia, se incluya un referente que facilite al desarrollador conocer los aspectos ergonómicos relevantes en la construcción de aplicaciones, de acuerdo a la especificaciones de los requisitos hechos por el usuario, así como el diseño y evaluación de las interfaces; de tal manera que se facilite la interacción entre éste y con el sistema.
Tabla 13: Análisis de la Norma ISO/TR 9241-100 (2010)	

ISO 9241-171: Orientación sobre accesibilidad de software			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input type="checkbox"/>	Norma <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	Se incluye en la investigación porque se considera importante que el nuevo estándar le permita al desarrollador, tener en cuenta en la construcción del software las condiciones físicas de las personas a la hora de usar el sistema, aportando así un valor agregado al mismo, elevando su calidad y competitividad.		
PROPÓSITO	Proporcionar la orientación y las especificaciones ergonómicas para el diseño de software de manera que éste sea accesible para su uso en el trabajo, en el hogar, en la educación y en lugares públicos. De igual manera, se ocupa de temas relacionados con el diseño de software accesible para personas con una amplia gama de capacidades físicas, sensoriales y cognitivas, incluidas las que están temporalmente discapacitadas y los ancianos.		
CONTENIDO	La norma contiene: <ul style="list-style-type: none"> ▪ Justificación y beneficios de la aplicación de accesibilidad. ▪ Principios para el diseño de software accesible. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO 9241-171: Orientación sobre accesibilidad de software	
CONTENIDO	<ul style="list-style-type: none"> ▪ Las fuentes de variación en las características del usuario. ▪ Lineamientos generales y requisitos, incluyendo la compatibilidad con la tecnología de asistencia. ▪ Entradas incluyendo teclados y dispositivos señaladores. ▪ Productos incluidos los resultados visual, auditivo y táctil. ▪ Documentación on-line, "Ayuda" y servicios de apoyo.
VENTAJAS	<ul style="list-style-type: none"> ▪ Es aplicable a la accesibilidad de los sistemas interactivos. ▪ Se dirige a una amplia gama de software (por ejemplo, de oficina, web, apoyo al aprendizaje y sistemas de biblioteca). ▪ Promueve el uso cada vez mayor de sistemas para una amplia gama de usuarios. ▪ Está destinado a ser utilizado por los responsables de la especificación, diseño, desarrollo, evaluación y obtención de las plataformas de software y aplicaciones de software.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ No se aplica a los requisitos para tecnologías de asistencia (assistive technologies) que se refieren a la utilización de tecnologías auxiliares como un componente integrado de los sistemas interactivos.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque se considera de gran importancia que la propuesta marco para Colombia, tenga unos parámetros que permitan evaluar la calidad del software en lo que a accesibilidad se refiere, teniendo en cuenta tanto el entorno donde se va a utilizar como las condiciones físicas de sus posibles usuarios.
Tabla 14: Análisis de la Norma ISO 9241-171 (2008)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

ISO/IEC 9000-3: Guía para la aplicación de la Norma ISO 9001 al Desarrollo de Software			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input checked="" type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	La ingeniería de software es una disciplina que tiene características propias diferentes a las de cualquier otro proceso de producción en general, por tal motivo, es importante conocer cuáles son los estándares para la gestión de la calidad en los procesos de producción de software propuestos por la norma ISO 9000 y como se deben aplicar.		
PROPÓSITO	Proporcionar una guía para las organizaciones en la aplicación de la norma ISO 9001:2000 a la adquisición, suministro, desarrollo, operación y mantenimiento de software y servicios de apoyo conexos.		
CONTENIDO	<p>La norma contiene:</p> <ul style="list-style-type: none"> ▪ Una serie de guías para las organizaciones en la aplicación de la norma ISO 9001:2000 en temas como nuevas notaciones, procedimientos y estándares. También se generaran documentos destinados a la dirección del grupo de desarrollo. ▪ Especificaciones de cómo aplicar la ISO 9001 a los procesos de software, entre ellos los procesos de adquisición, provisión, desarrollo, operación y mantenimiento y servicios de ayuda relacionados. ▪ Presenta temas como la Administración de la Configuración o Planeación de Proyectos que no están presentes en las normas de la familia ISO 9000 ▪ En las directrices propuestas en la norma ISO 9000-3 se tratan cuestiones tales como: <ul style="list-style-type: none"> • Comprensión de los requisitos funcionales entre el contratista y contratado. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC 9000-3: Guía para la aplicación de la Norma ISO 9001:2000 ⁸ al Desarrollo de Software	
VENTAJAS	<ul style="list-style-type: none"> ▪ Uso de metodologías coherentes para el desarrollo de software. ▪ Gestión de proyectos desde la concepción hasta el mantenimiento. ▪ No añade ni modifica de ninguna manera los requisitos de la norma ISO 9001:2000. ▪ Trata el concepto de ciclo de vida, pero sin imponer la utilización de uno en especial. ▪ Introduce otras actividades independientes a las fases del ciclo que son referentes a la configuración y distingue entre la verificación y la validación. ▪ No provee métodos específicos para evaluar la capacidad de aseguramiento de la calidad de una organización, motivo por el cual puede ser combinado con otros enfoques más específicos, como por ejemplo el modelo Espiral de Boehm.
DESVENTAJAS	Las directrices previstas en la norma ISO/IEC 90003:2004 no están destinadas a ser utilizadas como criterios de evaluación en el registro de sistema de gestión de calidad/certificación.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque se han encontrado en él una serie de deficiencias que han llevado a la revisión y mejora del mismo, el cual todavía se encuentra en proceso de cambio. Por otro lado, se considera que su aplicación proporciona una guía útil para detectar y corregir una serie de problemas de los productos software. Sin embargo, se espera que la nueva propuesta se base en algunos de los estándares mencionados anteriormente en el presente documento, lo que le permita mejorar la calidad de productos software sin necesidad de incluir en su contenido, al estándar en mención.
Tabla 15: Análisis del estándar ISO/IEC 90003 (2004)	

⁹ La norma ISO 9001:2000 especifica los requisitos para un sistema de gestión de calidad en una organización. International Organization for Standardization (ISO). (2000). *ISO 9001:2000*. Recuperado el 08 de Mayo de 2010, de http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21823

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC 12207: Sistemas e ingeniería de software – Procesos de ciclo de vida del software			
ORGANIZACIÓN	International Organization for Standardization (ISO)		
TIPO	Estándar <input checked="" type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se incluye en la investigación porque se considera importante conocer los procesos, actividades y tareas establecidas internacionalmente a manera de estándar, para la implementación de un modelo de ciclo de vida del software que permita desarrollar productos de calidad, cumpliendo a la vez con los plazos, costos y demás requerimientos establecidos tanto por el cliente como por el desarrollador.		
PROPÓSITO	Establecer un marco común para los procesos de ciclo de vida del software, con una terminología bien definida, que puede hacer referencia a la industria del software.		
CONTENIDO	Contiene los procesos, actividades y tareas que se deben aplicar durante la adquisición de un producto de software o servicio y durante el suministro, desarrollo, operación, mantenimiento y eliminación de productos de software.		
VENTAJAS	<ul style="list-style-type: none"> ▪ Se aplica a la adquisición, suministro, desarrollo, operación, mantenimiento y disposición de sistemas y productos de software y servicios. ▪ Proporciona un proceso que puede ser empleado para definir, controlar y mejorar los procesos de ciclo de vida del software. 		
DESVENTAJAS	<ul style="list-style-type: none"> ▪ La norma describe los procesos, pero no especifica como implementarlos. ▪ Enmarca las condiciones que debe cumplir el proceso de ingeniería de software a lo largo de todo el ciclo de vida del producto, desde el comienzo hasta el final de su vida útil, pero no detalla los bloques básicos para construir dicho proceso, como son las entradas y salidas que debe generar cada actividad. ▪ No incluye las guías sobre cómo interpretar cada actividad. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC 12207: Sistemas e ingeniería de software – Procesos de ciclo de vida del software	
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	El proceso de ciclo de vida del software es de vital importancia para el desarrollo de software, puesto que determina las estrategias, actividades, métodos y tareas, que debidamente organizadas permiten lograr un conjunto de metas y alcanzar los objetivos propuestos. Por tal motivo, se hace importante que en la creación de una nueva propuesta, se consideren los aspectos de ciclo de vida del software contenidos en normas tales como la ISO/IEC 12207 que pudieran llegar a ser relevantes en el proceso de desarrollo del software.
Tabla 16: Análisis de la norma ISO/IEC 12207 (2008)	

IEEE 1058.1: Planificación de gestión de proyectos de software			
ORGANIZACIÓN	Institute of Electrical and Electronics Engineers (IEEE)		
TIPO	Estándar <input checked="" type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se analiza en la investigación porque para muchas organizaciones el desarrollo de software va ligado a un plan de sistemas de información, y por ende, la gestión de proyectos juega un papel importante en el cumplimiento de las políticas y objetivos propuestos.		
PROPÓSITO	Especificar el formato y contenidos de los planes para la gestión de proyectos software.		
CONTENIDO	<p>El estándar está compuesto por tres secciones:</p> <ul style="list-style-type: none"> ▪ La Sección 1 define su alcance y proporciona referencias a otros estándares IEEE que deberían ser seguidos cuando se aplique el estándar en mención. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

IEEE 1058.1: Planificación de Gestión de Proyectos de Software	
CONTENIDO	<ul style="list-style-type: none"> ▪ La Sección 2 proporciona la definición de los términos que son usados a lo largo del estándar. ▪ La Sección 3 contiene una visión general y una especificación detallada del estándar, incluyendo los componentes requeridos que deben ser incluidos, y componentes adicionales que pueden ser incluidos en el plan del proyecto basado en este estándar.
VENTAJAS	<ul style="list-style-type: none"> ▪ El estándar identifica el conjunto mínimo de elementos que debería aparecer en todos los planes para la gestión de proyectos software. ▪ Su uso no está limitado por el tamaño, complejidad o criticidad del producto software. ▪ Este estándar es aplicable a todas las clases de software, incluyendo firmware, código de sistemas empotrados, matrices lógicas programables y software en Silicio. ▪ Puede ser aplicado a cualquiera de los segmentos del ciclo de vida de un producto software. ▪ Identifica el conjunto mínimo de elementos que debería aparecer en todos los planes para la gestión de proyectos software.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ No especifica las técnicas exactas que pueden ser usadas en el desarrollo de los planes de proyectos, ni ofrece ejemplos de los planes de gestión de proyectos. ▪ Cada organización que usa este estándar debería desarrollar un conjunto de prácticas y procedimientos para proporcionar una guía detallada para la preparación y actualización de los planes de gestión de los proyectos software basada en este estándar. ▪ Estas prácticas detalladas y procedimientos deberían tener en cuenta los factores del entorno, organizacionales y políticos que pueden influenciar en la aplicación de este estándar.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

IEEE 1058.1: Planificación de gestión de proyectos de software	
¿POR QUÉ?	<p>Porque al planificar la gestión de proyectos de software se puede hacer seguimiento y control a las actividades y recursos, tanto materiales como humanos, que intervienen en el desarrollo del producto software. Por este motivo, se considera importante que la nueva propuesta contemple incluir en su contenido la planificación de gestión de proyectos de software basada en el presente estándar, lo cual le permitirá conocer en todo momento qué problemas se pueden presentar y resolverlos lo más pronto posible, evitando trastornos en el tiempo y costo del proyecto de desarrollo del software.</p>
<p>Tabla 17: Análisis del estándar IEEE 1058.1 (1987)</p>	

8.2 Guías

SWEBOK Software Engineering Body of Knowledge							
ORGANIZACIÓN	Institute of Electrical and Electronics Engineers (IEEE)						
TIPO	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Estándar <input type="checkbox"/></td> <td style="padding: 5px;">Norma <input type="checkbox"/></td> <td style="padding: 5px;">Guía <input checked="" type="checkbox"/></td> </tr> <tr> <td style="padding: 5px;">Modelo <input type="checkbox"/></td> <td style="padding: 5px;">Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input checked="" type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input checked="" type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	<p>El ser una guía creada por una organización de carácter internacional como lo es la IEEE, en cuya elaboración participó un gran número de expertos, la convierte en un referente válido e importante a la hora de crear una nueva propuesta.</p>						
PROPÓSITO	<p>Dar una caracterización a la ingeniería del software como disciplina, para servir de guía en la construcción de software de calidad aplicando técnicas a nivel de ingeniería, de acuerdo a 10 áreas de conocimiento.</p>						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SWEBOK Software Engineering Body of Knowledge	
CONTENIDO	<p>La guía se subdivide en 10 áreas de conocimiento, en las que contempla una serie de aspectos y conceptos que llevan al software a ser objeto de aplicación de la ingeniería:</p> <ul style="list-style-type: none"> ▪ Requerimientos del software ▪ Diseño del software ▪ Construcción del software ▪ Pruebas del software ▪ Mantenimiento del software ▪ Gestión de la configuración del software ▪ Gestión de la ingeniería del software ▪ El proceso de ingeniería del software ▪ Herramientas y métodos de ingeniería del software ▪ Calidad del software
VENTAJAS	<ul style="list-style-type: none"> ▪ Al estar dividida en 10 áreas del conocimiento se contempla una serie de aspectos y conceptos que llevan al software a ser objeto de aplicación de la ingeniería, dando como resultado un producto de mayor calidad.
DESVENTAJAS	<p>Puede llegar a ser una guía de difícil aplicación debido al número de áreas que contempla.</p> <p>Al ser una guía aplicada a nivel mundial, es factible que se interprete de manera errónea y por ende, se aplique inadecuadamente en alguna organización</p>
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p>Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>
¿POR QUÉ?	<p>Porque es un cuerpo del conocimiento creado por expertos de diferentes áreas a nivel mundial, hecho que representa un referente confiable e importante a la hora de desarrollar una propuesta marco para Colombia.</p>

Tabla 18: Análisis de la Guía SWEBOK (2004)

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

8.3 Modelos de Desarrollo de Software

8.3.1 Modelos Clásicos

Modelo Lineal, Secuencial o en Cascada			
ORGANIZACIÓN	Propuesto inicialmente por Royce en 1970 y Boehm en 1981		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	Se analiza en la investigación porque se considera como el Modelo de Ciclo de Vida Básico, razón por la cual es uno de los más conocidos y aplicados en el medio.		
PROPÓSITO	Definir y seguir el progreso del desarrollo de software hacia puntos de revisión bien definidos, a través de un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.		
CONTENIDO	<p>Posee cinco etapas que se transforman en actividades fundamentales de desarrollo:</p> <ol style="list-style-type: none"> 1. Análisis y definición de requerimientos: Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. 2. Diseño del sistema y del software: El proceso de diseño del sistema divide los requerimientos en sistemas hardware y software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones. 3. Implementación y prueba de unidades: Durante esta etapa el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación. 4. Integración y prueba del sistema: Los programas o las unidades individuales de programas se integran y se prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo Lineal, Secuencial o en Cascada	
CONTENIDO	<p>5. Funcionamiento y mantenimiento: Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios una vez que se descubren nuevos requerimientos.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ La documentación se realiza en cada fase. ▪ Las metas se logran de mejor manera teniendo como fin puntos de revisión bien definidos y documentados, dividiendo el desarrollo en etapas secuenciales bien definidas.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Inflexibilidad a la hora de dividir el proyecto en varias etapas. ▪ No todos los requisitos son expuestos al principio de forma explícita, como lo requiere este modelo. ▪ El cliente debe tener paciencia, ya que una versión del programa no está disponible hasta que el proyecto esté muy avanzado. ▪ Un grave error puede ser desastroso si no se detecta hasta que se revisa el programa. ▪ Las iteraciones son costosas e implican rehacer trabajo debido a la producción y aprobación de documentos. ▪ Los problemas se dejan para su posterior resolución, lo que lleva a que estos sean ignorados o corregidos de una forma poco elegante. ▪ Es inflexible a la hora de evolucionar para incorporar nuevos requisitos. Es difícil responder a cambios en los requisitos.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p>Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>
¿POR QUÉ?	<p>Porque es uno de los modelos más conocidos y utilizados en el ámbito internacional para el desarrollo de software y contempla las fases principales del ciclo de vida del mismo.</p>

Tabla 19: Análisis del Modelo Lineal, Secuencial o Cascada

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo de Construcción de Prototipos			
ORGANIZACIÓN			
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se analiza en la investigación porque es un ciclo de vida clásico.		
PROPÓSITO	Desarrollar una implementación inicial para exponerla a los comentarios del usuario y posteriormente refinarla a través de las diferentes versiones hasta que se cree un sistema adecuado.		
CONTENIDO	El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un «diseño rápido». El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente (por ejemplo: enfoques de entrada y formatos de salida). El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.		
VENTAJAS	<ul style="list-style-type: none"> ▪ El usuario va viendo la evolución del sistema. ▪ Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios. 		
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Se desconoce el tiempo que se tardará en crear un producto aceptable, dado que la gestión de desarrollo de software es muy lenta. ▪ Se pueden adoptar prácticas de programación de prueba y error, sin realizar previamente un análisis y un diseño formal. ▪ Por el afán de hacer que funcione el prototipo, no se tiene en cuenta la calidad de software global o la facilidad para su mantenimiento a largo plazo. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo de Construcción de Prototipos	
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque es uno de los modelos más conocidos y utilizados en el ámbito internacional para el desarrollo de software.
Tabla 20: Análisis del Modelo de Construcción de Prototipos	

Modelo DRA (Desarrollo Rápido de Aplicaciones)							
ORGANIZACIÓN	Creado por James Martin en 1980						
TIPO	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input checked="" type="checkbox"/></td> <td>Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Se incluye en la investigación porque es un Modelo en el que se logra crear un sistema completamente funcional dentro de periodos cortos de tiempo.						
PROPÓSITO	Permitir al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo.						
CONTENIDO	<p>El modelo DRA es una adaptación a «alta velocidad» del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Este enfoque comprende las siguientes fases:</p> <p>1. Modelado de Gestión: El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?.</p> <p>2. Modelado de datos: El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.</p>						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

Modelo DRA (Desarrollo Rápido de Aplicaciones)	
CONTENIDO	<p>3. Modelado del proceso: Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos.</p> <p>4. Generación de aplicaciones: El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas para facilitar la construcción del software.</p> <p>5. Pruebas y entrega. Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Velocidad en el desarrollo. ▪ El DRA trabaja para volver a utilizar componentes de programas ya existentes o a crear componentes reutilizables.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado. Si no hay compromiso por ninguna de las partes constituyentes, los proyectos DRA fracasarán. ▪ No todos los tipos de aplicaciones son apropiados para DRA. Si un sistema no se puede modularizar adecuadamente, la construcción de los componentes necesarios para DRA será problemático. Si está en juego el alto rendimiento, y se va a conseguir el rendimiento convirtiendo interfaces en componentes de sistemas, el enfoque DRA puede que no funcione.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo DRA (Desarrollo Rápido de Aplicaciones)	
DESVENTAJAS	<ul style="list-style-type: none"> ▪ DRA no es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el software nuevo requiere un alto grado de interoperatividad con programas de computadora ya existentes.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p>Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>
¿POR QUÉ?	<p>Porque éste modelo es una adaptación a «alta velocidad» del modelo lineal secuencial que permite crear un sistema funcional dentro de periodos cortos de tiempo.</p>
<p>Tabla 21: Análisis del Modelo DRA (Desarrollo Rápido de Aplicaciones)</p>	

8.3.2 Modelos Evolutivos del Proceso de Software

Modelo Incremental			
ORGANIZACIÓN	Creado por Harlan Mills en 1980		
TIPO	<p>Estándar <input type="checkbox"/></p> <p>Modelo <input checked="" type="checkbox"/></p>	<p>Norma <input type="checkbox"/></p> <p>Metodología <input type="checkbox"/></p>	<p>Guía <input type="checkbox"/></p>
ANALIZADA EN LA INVESTIGACIÓN	<p>Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>		
JUSTIFICACIÓN	<p>Se incluye en la investigación porque es uno de los modelos más conocidos y utilizados para llevar a cabo el desarrollo de software.</p>		
PROPÓSITO	<p>Reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema.</p>		
CONTENIDO	<p>Es una combinación del Modelo en Cascada y el Modelo de Construcción de Prototipos. El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software.</p>		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

Modelo Incremental	
CONTENIDO	Requiere que un problema se divida en varios subproblemas para que cada cual se desarrolle a su vez. Según se completa cada sección, se verifica e integra con las demás secciones ya completadas del sistema. En cada paso, el sistema parcialmente completado se puede evaluar en relación al desarrollo de secciones futuras. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo.
VENTAJAS	<ul style="list-style-type: none"> ▪ Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento. ▪ Los clientes pueden aclarar los requisitos que no tengan claros conforme se van dando las entregas del sistema. ▪ Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento. ▪ Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Cada incremento debe ser pequeño para limitar el riesgo. ▪ Cada incremento debe aumentar la funcionalidad. ▪ Es difícil establecer las correspondencias de los requisitos contra los incrementos. ▪ Es difícil detectar las unidades o servicios genéricos para todo el sistema.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque es uno de los modelos más conocidos y utilizados en el ámbito internacional para el desarrollo de software, que permite la utilización del sistema sin estar completamente terminado.
Tabla 22: Análisis del Modelo Incremental	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo Espiral							
ORGANIZACIÓN	Creado por Barry Boehm en 1988						
TIPO	<table border="1" style="width: 100%;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input checked="" type="checkbox"/></td> <td>Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Se incluye en la investigación porque es uno de los modelos más conocidos y utilizados para llevar a cabo el desarrollo de software.						
PROPÓSITO	Proporcionar el potencial para el desarrollo rápido de versiones incrementales del software.						
CONTENIDO	<p>Es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales.</p> <p>El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas:</p> <p>Comunicación con el cliente: las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.</p> <p>Planificación: las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.</p> <p>Análisis de Riesgos: las tareas requeridas para evaluar riesgos técnicos y de gestión.</p> <p>Ingeniería: las tareas requeridas para construir una o más representaciones de la aplicación.</p> <p>Construcción y acción: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (por ejemplo: documentación y práctica).</p>						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo Espiral	
CONTENIDO	<p>Evaluación del Cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.</p> <p>Para proyectos pequeños, el número de tareas de trabajo y su formalidad es bajo. Para proyectos mayores y más críticos cada región de tareas contiene tareas de trabajo que se definen para lograr un nivel más alto de formalidad.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Una actividad comienza con un entendimiento de los objetivos y riesgos involucrados. ▪ Integra el desarrollo con el mantenimiento. ▪ Todo el personal relacionado debe involucrarse en una revisión que determina cada actividad, planeando y comprometiéndose con las siguientes actividades. ▪ Basado en la evaluación de soluciones alternas, se usa las herramientas que mejor reduzcan los riesgos. ▪ El desarrollo puede proceder en incrementos en cada etapa, permitiendo prototipos sucesivos del producto.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Requiere una considerable habilidad para la evaluación del riesgo, y se cuenta con esta habilidad para el éxito. ▪ Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas. ▪ El modelo no se ha utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos. ▪ Planificar un proyecto con esta metodología es a menudo imposible, debido a la incertidumbre en el número de iteraciones que serán necesarias.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p>Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>
¿POR QUÉ?	<p>Porque es un modelo conocido y utilizado a nivel internacional, que permite el desarrollo rápido de versiones incrementales del software.</p>

Tabla 23: Análisis del Modelo en Espiral

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo Espiral WIN-WIN			
ORGANIZACIÓN	Creado por Barry Boehm en 1998		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	No se incluye en la investigación porque es una variante del Modelo Espiral.		

Tabla 24: Análisis Modelo Espiral WIN-WIN

Modelo de Desarrollo Concurrente			
ORGANIZACIÓN	Propuesto por Davis y Sitaram en 1994		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se incluye en la investigación porque es uno de los modelos evolutivos más conocidos para llevar a cabo el desarrollo de software.		
PROPÓSITO	Define una serie de acontecimientos que dispararán transiciones de estado a estado para cada una de las actividades de la ingeniería del software.		
CONTENIDO	Se puede representar en forma de esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas. Es un modelo de tipo de red donde todas las personas actúan simultáneamente o al mismo tiempo. Este modelo se utiliza a menudo como el paradigma de desarrollo de aplicaciones cliente/servidor.		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo de Desarrollo Concurrente	
VENTAJAS	<ul style="list-style-type: none"> ▪ Es un buen modelo para proyectos en los que se conforman grupos de trabajo independientes. ▪ Proporciona una imagen exacta del estado actual de un proyecto.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Si no existen grupos de trabajo no se puede trabajar en este modelo.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque es un modelo aplicable como paradigma de desarrollo de aplicaciones cliente/servidor.
Tabla 25: Análisis del Modelo de Desarrollo Concurrente	

8.3.3 Desarrollo Basado en Componentes

Modelo de Desarrollo Basado en Componentes			
ORGANIZACIÓN			
TIPO	Estándar <input type="checkbox"/> Modelo <input checked="" type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input type="checkbox"/>	Guía <input type="checkbox"/>
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se analiza en la investigación porque es un modelo que conduce a la reutilización del software.		
PROPÓSITO	Configurar aplicaciones desde componentes preparados de software llamados clases (Reutilización de código elaborado con anterioridad)		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo de Desarrollo Basado en Componentes	
CONTENIDO	<p>Incorpora muchas de las características del modelo en espiral, es evolutivo por naturaleza y exige un enfoque iterativo para la creación del software. Configura aplicaciones desde componentes preparados de software. La actividad de la ingeniería comienza con la identificación de clases candidatas. Esto se lleva a cabo examinando los datos que se van a manejar por parte de la aplicación y el algoritmo que se va a aplicar para conseguir el tratamiento. Los datos y los algoritmos correspondientes se empaquetan en una clase. Las clases creadas en los proyectos de ingeniería del software anteriores, se almacenan en una biblioteca de clases o diccionario de datos.</p> <p>Una vez identificadas las clases candidatas, la biblioteca de clases se examina para determinar si estas clases ya existen. En caso de que así sea, se extraen de la biblioteca y se vuelven a utilizar. Si una clase candidata no reside en la biblioteca, se aplican los métodos orientados a objetos. Se compone así la primera iteración de la aplicación a construirse, mediante las clases extraídas de la biblioteca y las clases nuevas construidas para cumplir las necesidades únicas de la aplicación. El flujo del proceso vuelve a la espiral y volverá a introducir por último la iteración ensambladora de componentes a través de la actividad de ingeniería.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Conduce a la reutilización del software, ésta proporciona beneficios a los ingenieros encargados del desarrollo. ▪ Ayuda a reducir costos, tiempo y esfuerzo en el desarrollo del software. ▪ Ayuda a mejorar la fiabilidad, flexibilidad y la reutilización de la aplicación final. ▪ Simplifica las pruebas y el mantenimiento del sistema.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Requiere de mucho tiempo de trabajo adicional.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p>Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>
¿POR QUÉ?	<p>Porque es un modelo que facilita la reutilización del software facilitando en gran medida la labor del ingeniero.</p>
Tabla 26: Análisis del Modelo de Desarrollo Basado en Componentes	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

8.3.4 El Modelo de Métodos Formales

Modelo de Métodos Formales							
ORGANIZACIÓN	Propuesto por Robert Floyd en 1967 y posteriormente por Edsger Dijkstra en 1976						
TIPO	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input checked="" type="checkbox"/></td> <td>Metodología <input type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Se analiza en la investigación porque se considera de gran importancia conocer modelos diferentes a los clásicos utilizados tradicionalmente.						
PROPÓSITO	Especificar, desarrollar y verificar un sistema basado en computadora aplicando una notación rigurosa y matemática.						
CONTENIDO	<p>Comprende un conjunto de actividades que conducen a la especificación matemática del software de computadora.</p> <p>Los métodos formales se caracterizan por emplear técnicas y herramientas matemáticas que faciliten la construcción o el análisis de un modelo matemático de un sistema.</p>						
VENTAJAS	<ul style="list-style-type: none"> ▪ La ambigüedad, lo incompleto y la inconsistencia se descubren y se corrigen más fácilmente mediante la aplicación del análisis matemático. ▪ Cuando se utilizan métodos formales durante el diseño, sirven como base para la verificación de programas y por consiguiente permiten que el ingeniero del software descubra y corrija errores que no se pudieron detectar de otra manera. 						
DESVENTAJAS	<ul style="list-style-type: none"> ▪ El desarrollo de modelos formales actualmente es bastante caro y lleva mucho tiempo. ▪ Se requiere un estudio detallado porque pocos responsables del desarrollo de software tienen los antecedentes necesarios para aplicar métodos formales. ▪ Es difícil utilizar los modelos como un mecanismo de comunicación con clientes que no tienen muchos conocimientos técnicos. 						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo de Métodos Formales	
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque la utilización de métodos formales durante el desarrollo, proporciona un mecanismo para eliminar muchos de los problemas que son difíciles de superar con paradigmas de la ingeniería del software.
Tabla 27: Análisis del Modelo de Métodos Formales	

8.3.5 Modelos de Calidad del Software

CMM (Capability Maturity Model)			
ORGANIZACIÓN	SEI (Software Engineering Institute)		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se analiza en la investigación porque es un modelo conocido y aplicado mundialmente.		
PROPÓSITO	Evaluar los procesos en sus distintos niveles de madurez, identificar los niveles a través de los cuales una organización debe formarse para establecer una cultura de excelencia en la ingeniería de software.		
CONTENIDO	<p>CMM es un marco de trabajo que especifica guías para organizaciones de software que quieren incrementar su capacidad de procesos, considerando los siguientes puntos:</p> <ul style="list-style-type: none"> ▪ Identificar fortalezas y debilidades en la organización. ▪ Identificar los riesgos de seleccionar entre diferentes contratos y monitorear los mismos. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

CMM (Capability Maturity Model)	
CONTENIDO	<ul style="list-style-type: none"> Entender las actividades necesarias para planear e implementar los procesos de software. Ayudar a definir e implementar procesos de software en la organización a través de una guía. <p>Los procesos son evaluados a través de distintos niveles de madurez, que van desde prácticas desordenadas o <i>ad-hoc</i>, hasta lograr una mejora continua de procesos y por ende de producto. Éstos son: Nivel Inicial, Nivel Repetible, Nivel Definido, Nivel Administrado y Nivel Optimizado.</p>
VENTAJAS	<ul style="list-style-type: none"> Mejora la calidad en el desarrollo y mantenimiento de software. Provee una estructura conceptual para mejorar el manejo y el desarrollo de productos de software en una forma disciplinada y consistente.
DESVENTAJAS	<ul style="list-style-type: none"> Ha evolucionado a una versión más actualizada denominada CMMI (Capability Maturity Model Integration).
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p style="text-align: center;">Si <input type="checkbox"/> No <input checked="" type="checkbox"/></p>
¿POR QUÉ?	<p>Porque ya existe una nueva versión denominada CMMI (Capability Maturity Model Integration).</p>
Tabla 28: Análisis del Modelo CMM (Capability Maturity Model)	

CMMI (Capability Maturity Model Integration) Version 1.2: 2006			
ORGANIZACIÓN	Creado en SEI (Software Engineering Institute) y perteneciente a la Carnegie Mellon University		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	<p style="text-align: center;">Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>		
JUSTIFICACIÓN	Se analiza porque es una evolución del Modelo CMM		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

CMMI (Capability Maturity Model Integration) Version 1.2: 2006	
PROPÓSITO	Proponer un conjunto de mejores prácticas que pueden emplearse para evaluar y mejorar los procesos de una organización, proporcionándole los elementos esenciales para mejorar su rendimiento.
CONTENIDO	<p>Está estructurado de acuerdo a cinco niveles de madurez (Nivel 1: Inicial, Nivel 2: Administrado, Nivel 3: Definido, Nivel 4: Administrado Cuantitativamente y Nivel 5: Optimizado) en los que las organizaciones pueden ubicarse, dependiendo del grado de sofisticación de sus procesos. A su vez, cada nivel de madurez -con excepción del inicial- queda caracterizado por un conjunto de áreas de proceso que agrupan prácticas que, al ser ejecutadas colectivamente, permiten cumplir con algún objetivo que es considerado importante para el modelo.</p> <p>En CMMI cada nivel de madurez contiene varias áreas de proceso, las que a su vez quedan definidas por uno o varios objetivos específicos y un objetivo genérico. Cada uno de ellos tiene vinculado un conjunto de prácticas, llamadas específicas y genéricas respectivamente.</p> <p>Disciplinas: El modelo incluye cuatro cuerpos de conocimiento distintos: ingeniería de software, ingeniería de sistemas, desarrollo integrado de productos y procesos, y adquisición de productos. Si bien el modelo es esencialmente el mismo para las cuatro disciplinas, en algunas de ellas se agregan áreas de proceso o prácticas específicas.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ El ser una nueva generación de Modelos de Madurez como CMM y CMM-SW, lo convierten en un Modelo de amplia trayectoria para la evaluación y la mejora de los procesos de desarrollo de software en las organizaciones. ▪ Incluye cuatro cuerpos del conocimiento distintos, los cuales facilitan su aplicación en una disciplina determinada.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ CMMI es solamente un marco de referencia, razón por la cual podría darse una interpretación y/o aplicación errónea. ▪ CMMI apunta fundamentalmente a la eficiencia operativa. Por consiguiente, alcanzar un determinado nivel de madurez no garantiza el éxito de la organización.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	<p style="text-align: center;">Si <input checked="" type="checkbox"/> No <input type="checkbox"/></p>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

CMMI (Capability Maturity Model Integration) Versión 1.2:2006	
¿POR QUÉ?	Es importante tener en cuenta en la creación de una nueva propuesta marco para Colombia, la aplicación de Modelos de Madurez del Proceso de Producción de Software que ayuden a mejorar la calidad del producto y a estandarizar su producción.
Tabla 29: Análisis del Modelo CMMI Versión 1.2 (2006)	

Modelo de Procesos para la Industria del Software (MoProSoft)			
ORGANIZACIÓN	Secretaría de Economía de México		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	Se analiza porque es uno de los modelos creados en Latinoamérica.		
PROPÓSITO	Proporcionar a la industria de software en México un modelo basado en las mejores prácticas internacionales que sea fácil de entender, fácil de aplicar y a bajo costo; que permita elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.		
CONTENIDO	<p>El modelo de procesos (MoProSoft) considera nueve procesos distribuidos en tres categorías: Alta Dirección, Gerencia y Operación, que reflejan la estructura de una organización:</p> <p>Categoría de Alta Dirección: Aborda las prácticas de alta dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos. Contiene el proceso de Gestión del Negocio.</p>		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo de Procesos para la Industria del Software (MoProSoft)	
CONTENIDO	<p>Categoría de Gerencia: Esta categoría aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información por éstos y comunica los resultados a la Categoría de Alta Dirección. Los procesos pertenecientes a ésta categoría son: Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos (Recursos Humanos y Ambiente de Trabajo; Bienes, Servicios e Infraestructura, Conocimiento de la Organización).</p> <p>Categoría de Operación: Aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados. Los procesos pertenecientes a ésta categoría son: Administración de Proyectos Específicos y Desarrollo y Mantenimiento de Software.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Es un modelo enfocado hacia el desarrollo y mantenimiento de software. ▪ Fácil de entender. ▪ Es definido como un conjunto de procesos, lo que posibilita su aplicación en diferentes sectores de la organización. ▪ Práctico y fácil de aplicar, especialmente en organizaciones pequeñas. ▪ Es un modelo que además de servir como referencia para la certificación, contribuye al cumplimiento de los objetivos establecidos por la organización a través de la mejora de sus procesos.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Es un modelo enfocado hacia la industria mexicana. ▪ Es necesario contar con un mecanismo de evaluación, que indique un estado real de una organización durante un periodo de vigencia específico.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque a pesar de ser un modelo enfocado a la industria mexicana, podría servir de base en la creación de un nuevo modelo para Colombia.
Tabla 30: Análisis del Modelo de Procesos para la Industria del Software (MoProSoft)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo MPS (Mejora de Proceso del Software)			
ORGANIZACIÓN	Asociación para Promoción de la Excelencia del Software Brasileño (SOFTEX) - 2003		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Se analiza en la investigación porque es uno de los modelos creados en Latinoamérica.		
PROPÓSITO	Definir y perfeccionar un modelo de mejora y evaluación de proceso de software, dando preferencia a las micro, pequeñas y medianas empresas, de modo que atiendan sus necesidades de negocio y que sea reconocido nacional e internacionalmente como un modelo aplicable a la industria de software.		
CONTENIDO	<p>El modelo MPS establece un modelo de procesos de software y un método de evaluación de procesos. Esta estructura da sustentación al modelo MPS y asegura que esté siendo empleado de modo coherente con sus definiciones. El modelo MPS establece también un modelo de negocio para apoyar su adopción por las empresas desarrolladoras de software.</p> <p>El modelo MPS se basa en los conceptos de madurez y capacidad de proceso para la evaluación y mejora de la calidad y productividad de productos de software y servicios asociados. Dentro de ese contexto, el modelo MPS posee tres componentes: Modelo de Referencia (MR-MPS), Método de Evaluación (MA-MPS) y Modelo de Negocio (MN-MPS).</p> <p>La Guía de Evaluación contiene el proceso y el método de evaluación MA-MPS, los requisitos para los evaluadores líderes, evaluadores adjuntos e Instituciones Evaluadoras. El proceso y el método de evaluación MA-MPS están en conformidad con la Norma Internacional ISO/IEC 15504-2.</p> <p>El Modelo de Negocio MN-MPS describe reglas de negocio para implementación del MR-MPS por las Instituciones Implementadoras, evaluación siguiendo el MA-MPS por las Instituciones Evaluadoras, organización de grupos de empresas por las Instituciones Organizadoras de Grupos de Empresas (IOGE) para implementación del MR-MPS y evaluación MA-MPS, certificación de Consultores de Adquisición (CA) y programas anuales de entrenamiento del MPS.BR por medio de cursos, pruebas y workshops.</p>		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Modelo MPS (Mejora de Proceso del Software)	
CONTENIDO	El Modelo de Negocio MN-MPS describe reglas de negocio para implementación del MR-MPS por las Instituciones Implementadoras, evaluación siguiendo el MA-MPS por las Instituciones Evaluadoras, organización de grupos de empresas por las Instituciones Organizadoras de Grupos de Empresas (IOGE) para implementación del MR-MPS y evaluación MA-MPS, certificación de Consultores de Adquisición (CA) y programas anuales de entrenamiento del MPS.BR por medio de cursos, pruebas y workshops.
VENTAJAS	<ul style="list-style-type: none"> ▪ Es un Modelo de Procesos basado en normas internacionales (ISO/IEC 12207:2008, ISO/IEC 15504, CMMI-DEV). ▪ Incluye un Modelo de Negocio para apoyar la adopción del Modelo MPS en las empresas desarrolladoras de software. ▪ Puede ser aplicado a las micro, pequeñas y medianas empresas.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ El hecho de que sea un Modelo relativamente nuevo, lo hace poco conocido e implementado, razón por la cual no se cuenta con muchas evidencias de los resultados obtenidos de su aplicación en la industria. ▪ Es un modelo diseñado para atender las necesidades de las empresas de software del Brasil.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque está basado en normas y estándares reconocidos internacionalmente, además posee características que no están contempladas en otros modelos.
Tabla 31: Análisis del Modelo MPS (Mejora de Proceso del Software)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

ISO/IEC 15504: Software Process Improvement and Capability dEtermination (SPICE)			
ORGANIZACIÓN	International Organization for Standardization (ISO) e International Electrotechnical Commission (IEC) en 1993.		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input checked="" type="checkbox"/>	Metodología <input type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/>		No <input checked="" type="checkbox"/>
JUSTIFICACIÓN	<p>Porque inicialmente se derivó del ciclo de vida del proceso estándar ISO 12207 y de modelos de madurez como Trillium y CMM (analizados previamente en el presente documento).</p> <p>Además porque en una revisión al proyecto realizada en el 2004, el modelo fue eliminado y reemplazado por ISO 12207 (Procesos del ciclo de vida del software).</p>		
Tabla 32: Análisis del Modelo SPICE (1993)			

8.4 Metodologías

Los primeros desarrollos no tuvieron una metodología definida, fueron construidos de forma artesanal y se les llamó desarrollos convencionales. Éstos se caracterizaron por obtener resultados finales impredecibles, por no tener una forma de controlar lo que estaba sucediendo en el proyecto y porque los cambios organizativos afectaban negativamente al proceso de desarrollo. Estas limitaciones condujeron a una serie de problemas y por lo tanto a una búsqueda más sistemática en los procesos de construcción y mejora de la calidad del software, de la cual surgieron diferentes metodologías, agrupadas dentro de unas clasificaciones específicas, siendo las más conocidas: las Metodologías de Análisis y Diseño Estructurado conocido por sus siglas en inglés como SA/SD (Structured Analysis and Structured Design), las Metodologías Orientadas a Objetos y las Metodologías de Desarrollo Ágil. En la Tabla No.19 se describen algunas generalidades importantes de dichas metodologías. De igual forma, en la Tabla No.20 se muestran algunas de las principales metodologías estructuradas y orientadas a objetos, y en la Tabla No.21 se aprecian algunas convergencias y divergencias en la definición de las metodologías ágiles más importantes.

ORIENTACIÓN METODOLÓGICA	CARACTERÍSTICAS	HERRAMIENTAS/MODELADO
ESTRUCTURADA	<p><i>“Se basa primordialmente en la división entre funciones y datos” (Weitzenfeld, 2002, cap. 3, pág. 23).</i></p> <p><i>“Representan los procesos, flujos y estructuras de datos, de una manera jerárquica, descendente.</i></p> <ul style="list-style-type: none"> ▪ <i>Orientados a procesos: Se centran en la parte del proceso.</i> ▪ <i>Orientados a datos: Se orientan más a las entradas y salidas” (López Quesada, 2006).</i> 	<ol style="list-style-type: none"> 1. Diagramas de flujos de datos, que sirven para modelar la transformación de datos entre funciones del sistema. 2. Diagramas de transición de estados que sirven para modelar el comportamiento en el tiempo. Describen el efecto de eventos externos en los eventos y funciones. 3. Diagramas de entidad-relación, sirven para modelar el almacenamiento de los datos (Pinzón & Guevara Bolaños).

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORIENTACIÓN METODOLÓGICA	CARACTERÍSTICAS	HERRAMIENTAS/MODELADO
ORIENTADA A OBJETOS	<p>Se basa <i>“principalmente, en el modelo de un sistema en términos de objetos.</i></p> <p><i>A diferencia de las metodologías estructuradas, se identifican inicialmente los objetos del sistema, para luego especificar su comportamiento”</i> (Pinzón & Guevara Bolaños, pág. 95).</p> <p><i>“Ofrecen una forma de pensar más que una forma de programar.</i></p> <p><i>Reducen la complejidad en el diseño de software, y permiten atacar los errores durante el diseño en lugar de durante la implementación, donde el costo de reparación es bastante mayor”</i> (Weitzenfeld, 2002, cap. 3, pág. 23).</p> <p>Está basada en varias técnicas, incluyendo herencia, abstracción, modularidad, jerarquía, polimorfismo y encapsulamiento.</p>	<ol style="list-style-type: none"> 1. Diagramas de clase: Sirven para describir los componentes esenciales de la arquitectura de un sistema. 2. Diagramas de casos de uso: Especifican un sistema en término de su funcionalidad. 3. Diagramas de transición de estado: Describen los cambios de estado en los objetos, siendo equivalentes sus similares en las metodologías estructuradas. 4. Diagramas de secuencia: Sirven para describir los aspectos dinámicos del sistema, mostrando el flujo de eventos entre objetos en el tiempo. 5. Diagramas de colaboración: se utilizan para describir la comunicación entre objetos de un sistema. 6. Diagramas de subsistemas: se usan para describir agrupaciones de clases en un sistema (Pinzón & Guevara Bolaños).
DESARROLLO ÁGIL	<p>Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.</p> <p>Se hace mucho más importante crear un producto software que funcione que escribir mucha documentación.</p> <p>El cliente está en todo momento colaborando en el proyecto.</p> <p>Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan (Carrillo, Pérez & Rodríguez, (2008).</p> <p><i>“Especialmente preparadas para cambios durante el proyecto.</i></p> <p><i>Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.</i></p>	<p>Story Cards o Historias de Usuario, son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. (Muy usadas en XP).</p> <p>Tarjetas CRC (Clase, Responsabilidad y Colaboración), creadas por Kent Beck y Ward Cunningham en 1989 (Casas & Reinaga) y utilizadas inicialmente para el diseño de software orientado a objetos, facilitan la interacción entre los integrantes del proyecto en el modelado conceptual y/o diseño detallado del sistema a partir de la especificación de requisitos, las historias de usuarios o los casos de uso, especificando las responsabilidades y colaboraciones.</p>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ORIENTACIÓN METODOLÓGICA	CARACTERÍSTICAS	HERRAMIENTAS/MODELADO
DESARROLLO ÁGIL	<p><i>Entrega continua y en plazos breves de software funcional</i>" (Calderón, Valverde, & Jorge, 2007).</p> <p>Se basan en la creencia de que la parte más importante de la documentación es el código fuente y que la utilización de métodos y herramientas de modelado hacen más burocrático y lento el desarrollo software, limitando además la capacidad de respuesta ante los cambios.</p>	Herramientas de soporte para pruebas, tales como frameworks ¹⁰ de testing automático, como JUnit o cualquiera de sus versiones para diferentes lenguajes.

Tabla 33: Características generales de las metodologías estructuradas y orientadas a objetos

TIPO	METODOLOGÍA	FECHA	TIPO	METODOLOGIA	FECHA
Estructurada	Merise	1978	Orientada a objetos	SYNTHESIS	1989
	SSADM	1981		OOSD	1990
	SSADM Versión 3	1986		OMT	1991
	Métrica Versión inicial	1989		Fusión	1994
	SSADM Versión 4	1990		OOA/D	1994
	Métrica Versión 2	1993		MOSES	1994
	Métrica Versión 2.1	1995		Syntropy	1994
	Métrica Versión 3	2001		MEDEA	1994
			RUP	1998	

Tabla 34: Metodologías estructuras y orientadas a objetos

¹⁰ El término Framework hace referencia a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un Framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le pueden añadir las últimas piezas para construir una aplicación concreta. Gutiérrez, J. J. (s.f.). *¿Qué es un framework web?* Recuperado el 29 de Abril de 2010, de Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

METODOLOGÍA	ACRÓNIMO	CREACIÓN	TIPO DE MODELO	CARACTERÍSTICA
Adaptive Software Development	ASD	Highsmith 2000	Prácticas + ciclo de vida	Inspirado en sistemas adaptativos complejos.
Agile Modeling	AM	Amber 2002	Metodología basada en la práctica	Suministra modelado ágil a otros métodos.
Cristal Methods	CM	Cockbum 1998	Familia de metodologías	Metodología ágil con énfasis en modelo de ciclos.
Agile RUP	dX	Booch, Martin, Newkirk 1998	Framework/Disciplina	XP dado vuelta con artefactos RUP.
Dynamic Solutions Delivery Model	DSDM	Stapleton 1997	Framework/Modelo de ciclo de vida	Creado por 16 expertos en RAD.
Evolutionary Project Management	EVO	Gilb 1976	Framework adaptativo	Primer método ágil existente.
eXtreme Programming	XP	Beck 1999	Disciplina en prácticas de ingeniería	Método ágil radical.
Feature-Driven Development	FDD	De Luca & Coad 1998 Palmer & Felsing 2002	Metodología	Método ágil de diseño y construcción.
Lean Development	LD	Charette 2001, Mary y Tom Poppendieck	Forma de pensar - modelo lógico	Metodología basada en procesos productivos.
Rapid Development	RAD	McConnell 1996	Survey (estudio) de técnicas y modelos	Selección de <i>best practices</i> , no método
Microsoft Solutions Framework	MSF	Microsoft 1994	Lineamientos, disciplinas, prácticas	Framework de desarrollo de soluciones.
Scrum	Scrum	Sutherland 1994 Schwaber 1995	Proceso – Framework de management	Complemento de otros métodos, ágiles o no.

Tabla 35: Convergencias y divergencias entre las principales metodologías ágiles

(Calderón, Valverde Rebaza, & Jorge, 2007, págs. 11,12)

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Algunas de las Metodologías más representativas en la historia de la Ingeniería del Software son las siguientes:

AÑO	METODOLOGÍA
1968	Concepción sobre la programación estructurada de DIJKSTRA
1974	Técnicas de programación estructurada de WARNIER y JACKSON
1975	Primeros conceptos sobre diseño estructurado de MYERS y YOURDON
1977	Primeros conceptos sobre el análisis estructurado GANE y SARSON
1978	Análisis estructurado: DEMARCO y WIINBERG. Nace MERISE
1981	SSADM. Information Engineering
1985	Análisis y Diseño estructurado para sistemas de tiempo real de WARD y MELLOR
1986	SSADM Versión 3
1987	Análisis y Diseño estructurado para sistemas de tiempo real de HATLEY Y PIRHBAY
1989	METRICA
1990	SSADM Versión 4
1993	METRICA Versión 2
1995	METRICA Versión 2.1
2001	(METRICA Versión 3.0)

Tabla 36: Metodologías más representativas en la historia de la Ingeniería del Software

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

8.4.1 Metodologías Estructuradas

MERISE 1978							
ORGANIZACIÓN	Ministerio de Industria Francés. Desarrollada por Tardieu, Rochfeld y Colleti.						
TIPO	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	MERISE ha sido una de las principales metodologías para el análisis y diseño de sistemas de información desde 1976.						
PROPÓSITO	Facilitar el análisis, diseño, desarrollo e implementación de sistemas de información, aportando un plan de trabajo y técnicas de modelado para la concepción de aplicaciones coherentes para el área de gestión de empresas, suponiendo una intersección entre informática y organización.						
CONTENIDO	<p>Contiene las cuatro fases fundamentales del ciclo de vida del desarrollo software:</p> <ul style="list-style-type: none"> ▪ Estudio preliminar: En esta fase se analiza la situación existente y la propuesta de una solución global atendiendo a los criterios de gestión, de la organización y decisiones adoptadas por el comité directivo del proyecto. ▪ Estudio detallado: En esta fase se define a nivel funcional la solución. ▪ Implementación: Es la fase donde se describe el entorno técnico con la distribución de los datos en los ficheros y los tratamientos en módulos de programas, también se construyen los programas con su codificación y pruebas. ▪ Realización y puesta en marcha: En esta fase se realiza la instalación, formación del personal, así como la implantación de los medios técnicos y organizativos y la recepción por parte del usuario. <p>MERISE introduce dos ciclos complementarios: ciclo de abstracción y ciclo de decisión. El ciclo de abstracción se basa en tres niveles:</p> <ul style="list-style-type: none"> ▪ Nivel conceptual: Es donde se define el “qué” es decir, los objetivos y limitaciones. En este nivel se realiza un tratamiento de los datos según el modelo conceptual de datos y los procesos según el modelo conceptual de procesos. 						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MERISE 1978	
CONTENIDO	<ul style="list-style-type: none"> ▪ Nivel Organizativo: Es donde se define la organización adecuada que hay que implantar para alcanzar los objetivos asignados y se realiza un tratamiento de los datos y el modelo organizativo de tratamientos para la realización de los procesos. ▪ Nivel físico u operativo: Se realiza la integración de los medios técnicos necesarios para el proyecto, utiliza el modelo físico de datos para los datos y el modelo operativo de tratamiento para los procesos.
VENTAJAS	<ul style="list-style-type: none"> ▪ Cubre las fases fundamentales del ciclo de vida del desarrollo software: Estudio preliminar, Análisis o estudio detallado, Diseño, Implementación, Realización y puesta en marcha. ▪ Se basa en la separación de datos y procedimientos que se realizan en varios modelos físicos y conceptuales. ▪ Suministrar la documentación completa y precisa para poder mantener el sistema fácilmente. ▪ En el desarrollo de un proyecto intervienen tres grupos de trabajo: comité director, comité de usuarios, equipo de desarrollo; con el fin de garantizar el cumplimiento de los objetivos propuestos y la interpretación adecuada de los requerimientos de la organización. ▪ Utiliza técnicas de diseño y modelado tales como: Diagramas de flujo de datos (DFD), modelo conceptual de datos (entidad/relación), modelo lógico de datos, modelo conceptual de tratamientos, modelo organizativo de tratamientos, modelo operacional de tratamientos; con el fin de adquirir una mejor comprensión y dominio del sistema.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ No está diseñada para realizar el mantenimiento de las aplicaciones.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque es una metodología desarrollada específicamente para la industria francesa, de manera que tomarla como base en la construcción de una propuesta para ser implementación en el entorno colombiano requeriría de grandes cambios. Sin embargo, aspectos tales como los ciclos complementarios, la estructura y documentación de la metodología pueden tomarse como referencia de manera muy general.
Tabla 37: Análisis de la Metodología MERISE (1978)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SSADM Versión inicial: 1981 (Structured Systems Analysis and Design Method)			
ORGANIZACIÓN	Ministerio de Administración Pública del Reino Unido.		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	En el presente documento se analizará la misma metodología en una versión más reciente (SSADM Versión 4: 1990), la cual contiene avances y mejoras significativos con respecto a ésta versión de 1981.		
Tabla 38: Análisis de la Metodología SSADM Versión inicial (1981)			

SSADM Versión 3: 1986 (Structured Systems Analysis and Design Method)			
ORGANIZACIÓN	Ministerio de Administración Pública del Reino Unido.		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	En el presente documento se analizará la misma metodología en una versión más reciente (SSADM Versión 4: 1990), la cual contiene avances y mejoras significativos con respecto a ésta versión de 1986.		
Tabla 39: Análisis de la Metodología SSADM Versión 3 (1986)			

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SSADM Versión 4: 1990 (Structured Systems Analysis and Design Method)			
ORGANIZACIÓN	Ministerio de Administración Pública del Reino Unido. Desarrollada inicialmente por Learmonth y Burchett Management Systems (LBMS) a partir de 1980 y continuada por el Central Computing and Telecommunications Agency (CCTA)		
TIPO	Estándar <input type="checkbox"/> Modelo <input type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	<p>Es la metodología de aproximación en cascada para el desarrollo de sistemas de información considerada como la más completa de las metodologías estructuradas, por su garantía contrastada a lo largo de los años por los desarrolladores informáticos.</p> <p>Es la metodología estándar de desarrollo de proyectos del gobierno del Reino Unido y está constituida por un conjunto integrado de normas que busca cumplir todos los requisitos de un método de desarrollo de buenos sistemas, a través del Análisis de Sistemas estructurados y Metodologías de Diseño.</p>		
PROPÓSITO	<ul style="list-style-type: none"> ▪ Ser un enfoque de sistemas para el análisis y diseño de sistemas de información. ▪ Asegurar que los proyectos puedan continuar con éxito si una pérdida de personal se produce, sin un efecto perjudicial sobre el proyecto. ▪ Mejorar la forma en que los proyectos son controlados y administrados. ▪ Permitir un uso más eficaz de personal con experiencia y sin experiencia y de su desarrollo. ▪ Hacer posible los proyectos con el apoyo de herramientas informáticas, por ejemplo, sistemas de ingeniería de software asistidos por computador. ▪ Mejorar la comunicación entre los participantes de un proyecto para obtener mayor eficiencia. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SSADM Version 4: 1990 (Structured Systems Analysis and Design Method)	
CONTENIDO	<p>Consiste en una arquitectura de especificación de 3 esquemas, considerando 3 áreas del sistema de información:</p> <ul style="list-style-type: none"> ▪ Capa externa mediante la que los usuarios interactúan con el sistema. ▪ Diseño interno. <p>Modelo conceptual que representa los requerimientos de negocio y sobre los que se basa el diseño interno.</p> <p>Cubre tres fases fundamentales del ciclo de vida del desarrollo software: Estudio de viabilidad, Análisis y Diseño; cada una con las distintas tareas que lo componen.</p> <ul style="list-style-type: none"> ▪ Estudio de Viabilidad ▪ Investigación del entorno actual ▪ Opciones de negocio del sistema ▪ Definición de requerimientos ▪ Opciones de diseño técnico ▪ Diseño lógico ▪ Diseño físico
VENTAJAS	<ul style="list-style-type: none"> ▪ Es un estándar abierto, es decir, está disponible gratuitamente para su uso en la industria y muchas empresas ofrecen soporte y formación en herramientas CASE para el uso del mismo. ▪ Hace énfasis en los requisitos y en la participación de los usuarios. ▪ Define claramente el proceso de producción: qué hacer, cuándo y cómo. ▪ Máxima flexibilidad en herramientas y técnicas de implementación, puesto que utiliza herramientas comunes (CASE comerciales) para el modelado de los diagramas. ▪ Teóricamente permite planear, manejar y controlar bien un proyecto, con el fin de entregar el producto a tiempo y cumplir con los plazos establecidos. ▪ Está pensado para responder a cambios en el entorno empresarial, ya que tanto en el desarrollo del proyecto como en la documentación se tienen en cuenta los objetivos de negocio y las necesidades de las empresas, ofreciendo la posibilidad de adaptar la planificación del proyecto a las necesidades reales del negocio.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SSADM Version 4: 1990 (Structured Systems Analysis and Design Method)	
VENTAJAS	<ul style="list-style-type: none"> ▪ No requiere de habilidades muy especiales y se puede enseñar fácilmente al personal. ▪ Reduce la tasa de error de los Sistemas de Información mediante la definición de un determinado nivel de calidad desde el inicio y a través de la comprobación constante del sistema. ▪ Separa la lógica y el diseño de sistemas físicos, de tal manera que su implementación no requiere de nuevo hardware o software, lo que reduce los costos. ▪ Da la posibilidad de reutilizar ciertas técnicas y herramientas en otros proyectos, reduciendo el costo y el tiempo invertido en el análisis.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Proporciona un conjunto de procedimientos para llevar a cabo el análisis y diseño, pero no cubre aspectos como la planificación estratégica ni entra en la construcción del código. ▪ Pone especial énfasis en el análisis del sistema y su documentación, con lo cual puede generar un exceso de análisis que puede llevar mucho tiempo y elevar los costos, especialmente en los grandes sistemas.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Aunque es una metodología estándar desarrollada para la industria de software del Reino Unido, constituye una buena oportunidad para el desarrollo de sistemas de información de excelente calidad, ya que es considerada una de las metodologías más completas y aborda aspectos que podrían ser relevantes en la construcción de una nueva propuesta.
Tabla 40: Análisis de la Metodología SSADM Versión 4: 1990	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MÉTRICA Versión inicial: 1989							
ORGANIZACIÓN	Ministerio de Administraciones Públicas España (MAP)						
TIPO	<table border="1"> <tr> <td>Estándar <input type="checkbox"/></td> <td>Norma <input type="checkbox"/></td> <td>Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>						
JUSTIFICACIÓN	En el presente documento se analizará la misma metodología en una versión más reciente (MÉTRICA Versión 3: 2001), la cual contiene avances y mejoras significativos con respecto a ésta versión de 1989.						
Tabla 42: Análisis de la Metodología MÉTRICA Versión inicial (1989)							

MÉTRICA Versión 2: 1993							
ORGANIZACIÓN	Ministerio de Administraciones Públicas España (MAP)						
TIPO	<table border="1"> <tr> <td>Estándar <input type="checkbox"/></td> <td>Norma <input type="checkbox"/></td> <td>Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>						
JUSTIFICACIÓN	En el presente documento se analizará la misma metodología en una versión más reciente (MÉTRICA Versión 3: 2001), la cual contiene avances y mejoras significativos con respecto a ésta versión de 1993.						
Tabla 41: Análisis de la Metodología MÉTRICA Versión 2 (1993)							

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MÉTRICA Versión 2.1: 1995			
ORGANIZACIÓN	Ministerio de Administraciones Públicas España (MAP)		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/>		No <input checked="" type="checkbox"/>
JUSTIFICACIÓN	En el presente documento se analizará la misma metodología en una versión más reciente (MÉTRICA Versión 3: 2001), la cual contiene avances y mejoras significativos con respecto a ésta versión de 1995.		
Tabla 43: Análisis de la Metodología MÉTRICA Versión 2.1 (1995)			

MÉTRICA Versión 3: 2001			
ORGANIZACIÓN	Ministerio de Administraciones Públicas España (MAP)		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/>		No <input type="checkbox"/>
JUSTIFICACIÓN	<p>Esta metodología supone un conjunto de normas, técnicas y documentos para el desarrollo del software de diversa complejidad, tamaño y ámbito, en cuya elaboración se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos.</p> <p>Además, es una metodología que se ha ido adaptando a las distintas tecnologías que han ido surgiendo y así se han desarrollado varias versiones significativas, lo cual se convierte en una práctica muy interesante para la industria del software, y por supuesto, para los intereses de la presente investigación.</p>		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MÉTRICA Versión 3: 2001	
PROPÓSITO	<p>Ofrecer a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:</p> <ul style="list-style-type: none">▪ Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.▪ Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.▪ Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.▪ Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad.▪ Facilitar la operación, mantenimiento y uso de los productos software obtenidos.
CONTENIDO	<ul style="list-style-type: none">▪ Su estructura contempla una serie de procesos principales o Macroprocesos, a partir de los cuales se descompone cada uno de los procesos en actividades, y éstas a su vez en tareas.<ul style="list-style-type: none">➢ Planificación de sistemas de información➢ Desarrollo de sistemas de información<ul style="list-style-type: none">• Estudio de viabilidad del sistema (EVS)• Análisis del sistema de información (ASI)• Diseño del sistema de información (DSI)• Construcción del sistema de información (CSI)• Implantación y aceptación del sistema (IAS)➢ Mantenimiento de sistemas de información▪ Aporta un conjunto de procesos (interfaces) orientados a la organización y como apoyo a los procesos principales, entre los que se encuentran:<ul style="list-style-type: none">➢ Gestión de proyectos (GP)➢ Seguridad (SEG)➢ Aseguramiento de la calidad (CAL)

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MÉTRICA Versión 3: 2001	
CONTENIDO	<ul style="list-style-type: none"> ➤ Gestión de configuración (GC) ▪ Incluye un Plan General de Garantía de Calidad (PGGC).
VENTAJAS	<ul style="list-style-type: none"> ▪ Se han tenido en cuenta las diferentes tipos de desarrollo y tecnologías actuales (estructurado, orientación a objetos, reutilización, cliente/servidor, etc.). ▪ Facilita la operación, el mantenimiento y uso de productos software. ▪ Hace mucho énfasis en el uso de estándares de calidad e ingeniería del software. ▪ Permite controlar el sistema como producto global a lo largo de su creación, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores durante el mismo, lo que se traduce en un aumento de calidad del proceso de desarrollo y de mejora de la productividad en la organización. ▪ Permite conocer en todo momento qué problemas se producen y resolverlos lo más pronto posible, lo cual evitará desviaciones temporales y económicas. ▪ Permite incorporar a los sistemas de información mecanismos de seguridad adicionales a los que se proponen en la propia metodología. ▪ Proporciona un marco común de referencia para la definición y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Al haber sido concebida para abarcar el desarrollo completo de Sistemas de Información sea cual sea su complejidad y magnitud, deberá adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto, dificultando su aplicación. ▪ Es una metodología dirigida principalmente al proceso de desarrollo del software, de tal manera que sólo refleja los aspectos del Mantenimiento, correctivo y evolutivo, que tienen relación con el Proceso de Desarrollo y no de Mantenimiento de Sistemas de Información.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MÉTRICA Versión 3: 2001	
¿POR QUÉ?	<p>Porque a pesar de ser una propuesta desarrollada para suplir las necesidades del medio Europeo y en particular de España, es importante conocerla ya que sus aportes permitirán tener una visión más amplia de lo que se debería tener en cuenta a la hora de iniciar el planteamiento de una propuesta para Colombia, que permita garantizar la calidad del software a través de un proceso organizado, responsable y muy bien documentado.</p> <p>Por otro lado, constituye un referente muy importante puesto que en su elaboración se han tenido en cuenta diversos métodos de desarrollo, estándares de ingeniería del software y calidad, así como la experiencia de los usuarios de las versiones anteriores para solventar los problemas o deficiencias detectados, lo cual constituye un gran aporte a la nueva propuesta.</p>
Tabla 45: Análisis de Metodología MÉTRICA Versión 3 (2001)	

PRINCE2: 1989 (Projects IN Controlled Environments)			
ORGANIZACIÓN	Central Computer and Telecommunications Agency (CCTA) Agencia Central de Computación y Telecomunicaciones del Reino Unido		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	En el presente documento se analizará la misma metodología en una versión más reciente (PRINCE2: 2005), la cual contiene avances y mejoras significativos con respecto a ésta versión de 1989.		
Tabla 44: Análisis de la Metodología PRINCE1: 1989			

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

PRINCE2: 2005 (Projects IN Controlled Environments)			
ORGANIZACIÓN	Central Computer and Telecommunications Agency (CCTA) Agencia Central de Computación y Telecomunicaciones del Reino Unido		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	La gestión de proyectos de software también es una parte importante del desarrollo de sistemas de información. Por tal motivo, es importante conocer metodologías reconocidas internacionalmente que como ésta, tratan de conseguir que la organización realice los proyectos correctos, en el momento adecuado y por las razones correctas y justificadas.		
PROPÓSITO	Ofrecer una guía de dominio público para la aplicación de las mejores prácticas en la gestión de los proyectos.		
CONTENIDO	<p>Presenta una combinación de 8 procesos, 8 componentes y 3 técnicas:</p> <p>Procesos:</p> <ul style="list-style-type: none"> ▪ Starting UP a Project: surge la necesidad de realizar algo. ▪ Initiating a Project: inicia el proyecto con sus métricas. ▪ Directing a Project: administración del proyecto per se. ▪ Managing stage boundaries: manejo efectivo de las diferentes etapas. ▪ Controlling a Stage: midiendo la eficiencia del proyecto. ▪ Managing product delivery: garantizando la entrega de lo deseado. ▪ Closing a Project: cierre formal de un proyecto. ▪ Planning: planeación de todos los recursos involucrados. <p>Componentes:</p> <ul style="list-style-type: none"> ▪ Organisation: define la estructura organizacional del proyecto. ▪ Plans: define los pasos a seguir, los reportes de recursos, etc. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

PRINCE2: 2005 (P)Rojects IN Controlled Environments	
CONTENIDO	<ul style="list-style-type: none"> ▪ Controls: administración de los procesos. ▪ Business Case: define los beneficios del negocio. ▪ Quality Management: define y mide la calidad del proyecto. ▪ Configuration Management: define las características y cómo serán medidos los productos a entregar de acorde a sus especificaciones. ▪ Change Control: define el proceso y procedimiento a seguir si hay algún cambio. ▪ Management of Risk: define las variables a considerar y como medir los riesgos que deben tomarse en un proyecto. <p>Técnicas:</p> <ul style="list-style-type: none"> ▪ Product-Based Planning: define los productos a entregar de acuerdo a los requerimientos solicitados. ▪ Change Controls: garantiza someter a procesos toda la gerencia del proyecto basada en tener bajo control cualquier cambio que ocurra. ▪ Quality Reviews: ayuda a revisar los estándares ya existentes y también poder buscar nuevos que puedan ser aplicados. También involucra la correcta toma de decisiones del proyecto, el manejo de proveedores y el manejo de la información.
VENTAJAS	<ul style="list-style-type: none"> ▪ Es una metodología estructurada y flexible para hacer un manejo efectivo de la administración y gerencia de proyectos de cualquier tamaño, en cualquier área y haciendo un uso efectivo de todas las variables interrelacionadas. ▪ Se centra en el establecimiento claro del ciclo de vida, la definición y medición de productos de negocio, el suministro de un conjunto de actividades para conseguir los productos de negocio y el establecimiento de una estructura organizativa con responsabilidades bien definidas para poder gestionar el proyecto de forma óptima. ▪ Garantiza control en el inicio, desarrollo y cierre de un proyecto. ▪ Permite reducir drásticamente los riesgos de fracaso de los proyectos. ▪ Es una metodología estructurada y flexible para hacer un manejo efectivo de la administración y gerencia de proyectos de cualquier tamaño, en cualquier área y haciendo un uso efectivo de todas las variables interrelacionadas.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

PRINCE2: 2005 (PProjects IN Controlled Environments)	
VENTAJAS	<ul style="list-style-type: none"> ▪ Se centra en el establecimiento claro del ciclo de vida, la definición y medición de productos de negocio, el suministro de un conjunto de actividades para conseguir los productos de negocio y el establecimiento de una estructura organizativa con responsabilidades bien definidas para poder gestionar el proyecto de forma óptima. ▪ Garantiza control en el inicio, desarrollo y cierre de un proyecto. ▪ Permite reducir drásticamente los riesgos de fracaso de los proyectos. ▪ Cubre todos los aspectos de organización, gestión y control de los proyectos, con el fin de lograr que los productos entregados lo sean en el tiempo establecido y con el presupuesto acordado. ▪ La metodología se puede aplicar a cualquier tipo de proyecto, y permite la gestión de los riesgos, el control de la calidad y la eficiencia de los cambios.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Es una metodología y no la receta de la garantía de éxito de un proyecto. ▪ No cubre todos los aspectos de la gestión de los proyectos. Hay ciertos aspectos propios de la gestión de los proyectos que no están contemplados en la metodología como pueden ser el liderazgo, las habilidades para la gestión de recursos, así como la cobertura detallada de técnicas y herramientas propias de la gestión.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque se considera que la metodología Métrica Versión 3: 2001 analizada en el presente documento, presenta un conjunto de procesos (interfaces) orientados a la Gestión de proyectos lo suficientemente completo como para manejar adecuadamente dicho aspecto. Sin embargo, es indudable que la metodología PRINCE2: 2005, por estar orientada exclusivamente a la gestión de proyectos constituye una muy buena alternativa para la aplicación de las mejores prácticas referentes al tema en mención.
Tabla 46: Análisis de la Metodología PRINCE2 (2005)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MAGERIT Versión 1.0 Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información			
ORGANIZACIÓN	Consejo Superior de Administración Electrónica (CSAE) de España		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	Existe una versión más reciente de la metodología (MAGERIT Versión 2.0: 2006).		
Tabla 48: Análisis de la Metodología MAGERIT Versión 1.0			

MAGERIT Versión 2.0: 2006 Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información			
ORGANIZACIÓN	Consejo Superior de Administración Electrónica (CSAE) de España		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	Porqué no es exactamente una metodología de desarrollo de software, sino que está directamente relacionada con la generalización del uso de las TI, ya que tiene la finalidad de proporcionar los principios básicos y requisitos mínimos para la protección adecuada de la información mediante la implantación y aplicación de un esquema de seguridad. Además, ésta metodología se ha tenido en cuenta en la creación de MÉTRICA Versión 3 (analizada anteriormente en el presente documento), por lo cual no se considera necesario incluirla en el análisis de la investigación, ya que muchos de sus aspectos han sido incluidos en la misma.		
Tabla 47: Análisis de la Metodología MAGERIT Versión 2.0 (2006)			

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

EUROMÉTODO Versión 1: 1998			
ORGANIZACIÓN	Comisión Europea (DG III, Regulatory policy and standardisation, electronic commerce, telematics networks, environment) auspiciado por el PPG, Grupo de Compras Públicas y publicado por el Ministerio de Administración Pública de España (MAP)		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Es una metodología ampliamente usada por organizaciones públicas y privadas en Europa para la adquisición de sistemas de información. Además fue tomada en cuenta en la creación de la metodología MÉTRICA Versión 3.		
PROPÓSITO	<ul style="list-style-type: none"> ▪ Ayudar a las organizaciones en la adquisición de sistemas de información efectivos y de los servicios asociados en distintas situaciones. ▪ Animar a los clientes y proveedores a controlar los costes y los plazos previstos, gestionar los riesgos y mejorar el entendimiento mutuo. ▪ Proporcionar un marco para la unificación de la terminología de los métodos. 		
CONTENIDO	<ul style="list-style-type: none"> ▪ Como marco metodológico proporciona un conjunto de conceptos y una terminología para mejorar la relación cliente-proveedor. ▪ Proporciona un marco en el cual describir, utilizar, perfeccionar, unificar y armonizar los distintos métodos, como por ejemplo los métodos de reingeniería de procesos de negocio, métodos de desarrollo de sistemas de información y métodos de gestión de proyectos y servicios. ▪ Proporciona un método para definir, planificar y ejecutar la adquisición de un sistema de información y los servicios asociados. 		
VENTAJAS	<ul style="list-style-type: none"> ▪ Es una metodología de carácter público que puede utilizarse libremente. Sus características de diseño permiten su utilización tanto en el ámbito público como en el privado. ▪ Puede utilizarse como marco metodológico y como método propiamente dicho. 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

EUROMÉTODO Versión 1: 1998	
VENTAJAS	<ul style="list-style-type: none"> ▪ Formulación más clara de los requisitos y mejor comprensión de las necesidades del cliente. ▪ Permite mejorar de la gestión de riesgos. ▪ Determinación del enfoque más adecuado para la realización de un proyecto o la prestación de un servicio. ▪ Selección de los métodos, técnicas y herramientas adecuadas.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ No se ocupa de aspectos legales ni es un método de desarrollo de software. ▪ Como metodología sólo proporciona un marco (teórico), es decir, un conjunto de conceptos y una terminología para mejorar la relación cliente-proveedor y unificar los métodos.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque es una metodología usada para la adquisición de sistemas de información más no proporciona prácticas de ingeniería de software para su construcción. Además, ha sido tomada en cuenta en la creación de MÉTRICA Versión 3, por lo cual no se considera necesario incluirla en el análisis de la investigación, ya que muchos de sus aspectos han sido incluidos en la misma.
Tabla 49: Análisis de la Metodología EUROMÉTODO Versión 1 (1998)	

8.4.2 Metodologías Orientadas a Objetos

RUP Rational Unified Process: 1998			
ORGANIZACIÓN	IBM Rational Software Fue creado por Grady Booch (Rational Software Corporation), Ivar Jacobson (Objectory) y James Rumbaugh (General Electric)		
TIPO	Estándar <input type="checkbox"/> Modelo <input type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

RUP Rational Unified Process: 1998	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
JUSTIFICACIÓN	Es una de las llamadas " <i>metodologías pesadas</i> " más conocidas y utilizadas para el desarrollo de proyectos software orientados a objetos.
PROPÓSITO	Facilitar, junto con UML, el desarrollo de software a través del análisis, la implementación y documentación de sistemas orientados a objetos.
CONTENIDO	<p>RUP tiene tres características esenciales:</p> <ul style="list-style-type: none"> ▪ Está dirigido por los Casos de Uso: Los cuales representan los requisitos funcionales del sistema, guían su diseño, implementación y prueba. ▪ Está centrado en la arquitectura: Involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. ▪ Es iterativo e incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un miniproyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. <p>La estructura estática del proceso unificado se define en base a cuatro elementos:</p> <ul style="list-style-type: none"> ▪ Roles: Definen el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. ▪ Actividades: Son unidades de trabajo que una persona que desempeñe ese rol puede realizar. Por lo regular consisten en crear o actualizar algún producto. ▪ Productos o artefactos: Son trozos de información producidos, modificados o usados por un proceso. Son los resultados tangibles del proyecto. ▪ Flujos de trabajo: Definen la secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos, que producen resultados observables.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

RUP Rational Unified Process: 1998	
CONTENIDO	<p>RUP se divide en cuatro fases:</p> <ul style="list-style-type: none">▪ Inicio: Explorar el problema y estimar los requisitos.▪ Elaboración: Analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos▪ Construcción: Alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones.▪ Transición: Alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y usabilidad del producto. <p>RUP identifica 6 <i>best practices</i> con las que define una forma efectiva de trabajar para los equipos de desarrollo de software:</p> <ul style="list-style-type: none">▪ Gestión de requisitos: Encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones.▪ Desarrollo de software iterativo: Desarrollo del producto mediante iteraciones con hitos bien definidos, en las cuales se repiten las actividades pero con distinto énfasis, según la fase del proyecto.▪ Desarrollo basado en componentes: La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema.▪ Modelado visual (usando UML): Para visualizar, especificar, construir y documentar los artefactos de un sistema software.▪ Verificación continua de la calidad: Las pruebas se integran a lo largo de todo el proceso para evaluar la calidad de todos los artefactos en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración.▪ Verificación continua de la calidad: Las pruebas se integran a lo largo de todo el proceso para evaluar la calidad de todos los artefactos en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

RUP Rational Unified Process: 1998	
CONTENIDO	<ul style="list-style-type: none"> ▪ Gestión de los cambios: Puesto que el cambio es un factor de riesgo crítico en los proyectos de software, es necesario controlar los cambios debidos tanto a acciones de mantenimiento posteriores a la entrega del producto, como a los producidos durante el proceso de desarrollo, especialmente importantes por su posible impacto son los cambios en los requisitos.
VENTAJAS	<ul style="list-style-type: none"> ▪ Está soportado en el Lenguaje de Modelamiento Unificado (UML), el cual es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, generalmente con orientación a objeto. ▪ Integra herramientas CASE y herramientas de asistencia al proceso, tales como aseguramiento de calidad y administración de requisitos, entre otros. ▪ Establece que es posible adecuar las actividades y reglas como mejor acomode a los desarrolladores, es decir, es adaptable al gusto del desarrollador. ▪ Es iterativo e incremental, centrado en la arquitectura (subsistemas, clases, componentes y nodos de los que se compone y las interfaces entre ellos, junto con su comportamiento) y dirigido por los casos de uso (que incluyen las pruebas) y los riesgos. ▪ Se adapta a las características de la organización para la que se esta desarrollando el software. ▪ Permite la colaboración entre los equipos a través de una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes y resultados, además de asignar tareas y responsabilidades. ▪ Motiva el uso de la reutilización. ▪ La calidad del producto se verifica en cada aspecto de la producción.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ El ser adaptable al gusto del desarrollador puede convertirse en un problema, puesto que en ocasiones se cree necesario establecer con claridad qué hacer, disponiendo de una enorme cantidad de opciones para decidir que realizar. ▪ Es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. ▪ En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

RUP Rational Unified Process: 1998	
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	<p>Porque es una de las metodologías más conocidas y usadas para el desarrollo de sistemas orientados a objetos, está soportada por UML (también muy conocido) y fue desarrollada por una compañía de gran trayectoria en el campo de la computación como lo es IBM. Además, es importante que la nueva metodología considere el desarrollo de sistemas orientados a objetos y pueda apoyarse en aspectos presentados en RUP que pueden ser bastante significativos.</p>
Tabla 50: Análisis de la Metodología RUP (1998)	

OORAM Object Oriented Role Analysis and Modelling: 1996							
ORGANIZACIÓN	Creada por Trygve Reenskaug						
TIPO	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	<p>Porque es un método de análisis y diseño basado en la orientación a objetos pero que introduce el concepto de modelo de roles como principal mecanismo de abstracción que utilizará el modelador, lo cual es importante y se considera debe ser tenido en cuenta.</p>						
PROPÓSITO	<p>Modelar grandes sistemas y favorecer una implementación en lenguajes de programación orientados a objetos, mediante el modelado con roles.</p> <p>También intenta desarrollar un marco de trabajo para la creación de distintas metodologías.</p>						

OORAM Object Oriented Role Analysis and Modelling: 1996	
CONTENIDO	<p>Como metodología:</p> <p>El sistema se descompone en un conjunto de subsistemas o áreas de interés que representan actividades desempeñadas por una estructura de objetos que colaboran entre sí. Cada una de estas estructuras es descrita mediante un modelo de roles.</p> <p>Un modelo de roles describe los objetos que participan en una actividad y las interacciones entre ellos; contiene un conjunto de roles, de modo que todos los objetos que ocupan una misma posición en la estructura son representados por un rol.</p> <p>Un rol describe el comportamiento de un objeto en el contexto de una actividad.</p> <p>Presenta su propio lenguaje de modelado, en el que se contemplan los siguientes modelos para representar los subsistemas o áreas de interés:</p> <ul style="list-style-type: none">▪ Área de interés: Es una descripción textual de la actividad a modelar.▪ Estímulo-respuesta: Muestra los roles del entorno que pueden tanto iniciar la actividad en el sistema enviando un mensaje estímulo, como recibir el resultado producido como respuesta a la actividad.▪ Colaboración: Representa los roles de un modelo junto con las conexiones que definen al interactuar.▪ Interfaz: Definición de la relación de mensajes susceptibles de ser enviados.▪ Escenario: Es similar al diagrama de secuencia definido por UML, con la salvedad de que en este caso lo que se representa no son mensajes entre instancias de clases, sino mensajes entre jugadores de roles.▪ Proceso: Describe los flujos de datos entre los roles y las actividades tomadas por los roles para procesar los datos.▪ Semántica: Vista similar a la de colaboración, donde las conexiones entre los roles implican relaciones de asociación. Describe los posibles estados de un rol, y los mensajes o eventos que disparan las transiciones de estado.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

OORAM Object Oriented Role Analysis and Modelling: 1996	
CONTENIDO	<p>Como marco para el desarrollo de metodologías:</p> <p>Se basa en una <i>arquitectura de tres modelos</i> para el modelado de sistema de información de gestión, así:</p> <ul style="list-style-type: none"> ▪ Modelo de empresa: En el que se identifican los distintos roles que juegan las personas que forman el dominio de la empresa, y la forma en la que colaboran para llevar a cabo las tareas dentro de la misma. ▪ Modelo de información: Centrado en la información que maneja la empresa y en como esta relacionada. ▪ Modelo de Tarea/Herramienta/Servicio: Describe las interfaces entre usuarios y servicios de información.
VENTAJAS	<ul style="list-style-type: none"> ▪ El concepto de rol unifica los conceptos clase y objeto: los roles tienen tanto una naturaleza estática como dinámica, pues permiten describir las propiedades de los objetos que representan, y también pueden usarse para mostrar cómo los objetos colaboran entre sí. ▪ Al igual que un objeto, un rol tiene identidad: puede enviar y recibir mensajes. ▪ Una clase describe un objeto independientemente del contexto en que dicho objeto interactúa con otros; un rol describe un objeto en el contexto de una actividad. ▪ Aporta su propio lenguaje de modelado, cuyas vistas son muy cercanas a las manejadas por los lenguajes de modelado orientados a objetos, como UML. ▪ Permite la construcción de modelos complejos a partir de otros más simples, facilitando a la vez la aplicación sistemática de componentes reutilizables. ▪ El modelo de proceso al que induce es mayormente el clásico en cascada, pero no presenta ninguna incompatibilidad con otros modelos.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ No contempla los conceptos de clase y herencia durante las actividades de análisis y desarrollo. ▪ La actividad de diseño no se encuentra estrictamente definida y queda abierta a las distintas metodologías que se creen dentro del marco de trabajo. ▪ El responsable del modelado debe moverse en un territorio más abstracto que el de la orientación a objetos (clases, objetos), puesto que todo el proceso de modelado del sistema se articula en torno a los roles y las actividades entre ellos.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

OORAM Object Oriented Role Analysis and Modelling: 1996	
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque además de modelar grandes sistemas y favorecer su implementación mediante roles, desarrollar un marco de trabajo para la creación de metodologías que podría contribuir significativamente en el planteamiento de una nueva propuesta.
Tabla 51: Análisis de la Metodología OORAM (1996)	

OOSE Object-Oriented Software Engineering: 1992			
ORGANIZACIÓN	Creada por Ivar Jacobson		
TIPO	Estándar <input type="checkbox"/> Modelo <input type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Porque junto con OMT (Object Modeling Technique de Rumbaugh) y el Método de Booch (G. Booch), es uno de los precursores del Lenguaje Unificado de Modelado (UML) tan usado actualmente para visualizar, especificar, construir y documentar los modelos de un sistema que involucra una gran cantidad de software, desde una perspectiva de Orientación a Objetos.		
PROPÓSITO	Proporcionar un soporte para el diseño creativo de productos de software, inclusive a escala industrial. Aportar un enfoque para el manejo de casos de uso, de manera que dicho modelo sirva como un modelo central del cual todos los otros modelos son derivados. Esto se da porque un modelo de casos de uso describe la funcionalidad completa del sistema, identificando como, todo lo que esta fuera del sistema, interactúa con él.		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

OOSE Object-Oriented Software Engineering: 1992	
CONTENIDO	<p>El modelo de casos de uso de acuerdo con Jacobson, es la base en la etapa de análisis, construcción y prueba. Por tal motivo, OOSE presenta cinco técnicas para modelar un sistema:</p> <ul style="list-style-type: none"> ▪ Modelo de requerimientos: Delimita el sistema y define su funcionalidad a través de la captura de requerimientos. ▪ Modelo de análisis: Estructura el sistema, modelando tres tipos de objetos (objetos de interface, objetos entidad y objetos de control), con el fin de dar al sistema una estructura de objetos robusta y flexible a los cambios. ▪ Modelo de diseño: Refina el modelo de análisis y lo adapta a un ambiente de implementación. Consiste de diagramas de interacción y diagramas de transición de estados. ▪ Modelo de implementación: Consiste en el código fuente de los objetos especificados en el modelo de diseño (implementación del sistema). ▪ Modelo de prueba: Es llevado acabo mediante la realización de pruebas al modelo de implementación (verificar el sistema). <p>La idea básica de estos modelos es capturar el concepto inicial de todos los requerimientos funcionales y usar sus perspectivas. Es por eso que la relación entre ellos es importante. Para hacer posible el mantenimiento del sistema es también necesario que los modelos sean tangibles.</p> <p>Durante el desarrollo de un sistema se crea un modelo del sistema. El diseño de este modelo trabaja sobre un proceso de descripción con el cual será desarrollado el sistema. Esos modelos son expresados en un cierto espacio de información. Cada proceso toma uno o varios modelos y transforma esto dentro de otros modelos.</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Primera metodología de diseño orientado a objetos que emplea los casos de uso en diseño de software. ▪ Todo el desarrollo del sistema es guiado por los casos de uso. ▪ Buen soporte de Ingeniería de Requisitos e Ingeniería de Información. ▪ Modelado y simulación de sistemas de telecomunicaciones. ▪ Por ser una de las fuentes primarias de UML, sus conceptos y notaciones se han incorporado a la misma.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

OOSE Object-Oriented Software Engineering: 1992	
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Las herramientas de OOSE han sido sustituidas por herramientas de apoyo a UML y RUP. ▪ Parte de la metodología ha evolucionado también en el Proceso Unificado de Rational (RUP).
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque en la actualidad OOSE ha sido reemplazado en gran medida por la notación UML y la metodología RUP, de manera que sus conceptos y notaciones ahora hacen parte de los mismos.
Tabla 52: Análisis de la Metodología OOSE (1992)	

OMT Object Modeling Technique: 1991			
ORGANIZACIÓN	Creada por James Rumbaugh y Michael Blaha		
TIPO	Estándar <input type="checkbox"/> Modelo <input type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input checked="" type="checkbox"/>	Guía <input type="checkbox"/>
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	Porque a pesar de ser una de las primeras metodologías de análisis y diseño orientadas a objetos, que presenta gran madurez y eficiencia, además de su carácter abierto (no propietaria); OMT fue una de las metodologías que junto con OOSE (Object Oriented Software Engineering) de Jacobson y OOAD (Object-Oriented Analysis and Design) de Grady Booch, evolucionó hasta formar el Método Unificado de Desarrollo de Software (RUP) presentado por IBM Rational Software, analizado anteriormente en el presente documento.		
Tabla 52: Análisis de la Metodología OMT (1991)			

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

OOAD Object-Oriented Analysis and Design: 1991			
ORGANIZACIÓN	Creada por Grady Booch		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	<p>Porque a pesar de que ha sido durante mucho tiempo la referencia esencial a la tecnología orientada a objetos, también ha evolucionado junto con otras metodologías como OMT (Object Modeling Technique) de James Rumbaugh & Michael Blaha y OOSE (Object Oriented Software Engineering) de Jacobson, hasta formar el Método Unificado de Desarrollo de Software (RUP) presentado por IBM Rational Software, analizado anteriormente en el presente documento. Su análisis no se incluye en la investigación, ya que se presume que los aspectos principales de la misma, hacen parte de la metodología RUP.</p>		
Tabla 54: Análisis de la Metodología OOAD (1991)			

SEOO Ingeniería de Sistemas Orientada a Objetos			
ORGANIZACIÓN	Compañía LBMS de U.K		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>		
JUSTIFICACIÓN	<p>Porque es una metodología propietaria poco mencionada, razón por la cual no hay mucha información detallada disponible sobre la misma.</p>		
Tabla 53: Análisis de la Metodología SEOO			

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

8.4.2.1 Otras Metodologías Orientadas a Objetos.

METODOLOGÍA/AUTOR	ACRÓNIMO	METODOLOGÍA/AUTOR	ACRÓNIMO
Business Object Notation, Nerson	BON	Object-Oriented Technology for Real-Time Systems, Maher Awad, Juha Kuusela y Jurgen Ziegler, 1993	Octopus
Catalysis, Desmond D'Souza y Addison Wesley, 1998	Catalysis	Object Oriented Structured Design, Wasserman, 1990	OOSD
Coad, Peter Coad y Edward Yourdon	Coad	Object-oriented Process, Environment and Notation, Brian Henderson-Sellers	OPEN / OML
Demeter, Karl J. Lieberherr, 1996	Demeter	Object Oriented Design OSA, Paul Black, Embley y otros	OSA
Design of OO Real-time Systems, Ray Buhr, 1992-1994	DOORS	Skill-Driven Design, Avner Ben	Skill-Driven Design
Designing OO Software, Rebecca Wirfs-Brock, Brian Wilkerson y Lauren Wiener	DOOS	Object COMX, Judith Barnard	Object COMX
The EROOS Method, Katholieke Universiteit Leuven	EROOS	SYNTHESIS, 1989	SYNTHESIS
Fusion, Derek Coleman, 1994	Fusión	SYNTROPY, Cook y otros, 1994	SYNTROPY
The Hierarchical OO Design method, ESA/ESTEC/WME	HOOD	Object-Oriented Design, Grady Booch	OOD
ION, Colin Atkinson	ION	Object Oriented Analysis, Coad y Yourdon	OOA
MERODE, Monique Snoeck	MERODE	Object Oriented Systems Analysis, Shaler y Mellor	OOSA
MEDEA, 1994	MEDEA	Responsibility Driven Design, Wirfs-Brock	RDD
Software Engineering of Systems, Henderson-Sellers & Edwards, 1994	MOSES	Objectory, Ivar Hjalmar Jacobson, 1993	Objectory
Object Behaviour Analysis, Rubin & Goldberg	OBA	Methodology for Object Oriented	MOO
General Object Oriented Design, Seidewitz & Stark	GOOD	Visual Modeling Technique	VMT IBM
Object Oriented Jackson, Jackson	OOJSD	ADM3, Donald Firesmith	ADM3
Colbert, E. Colbert	Colbert	Frame Object Analysis, Andleigh/Gretzingr	FOA
Semantic Object Modelling Approach, Ian Graham	SOMA	Ptech, Martin & Odell	OOA&D

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

METODOLOGÍA/AUTOR	ACRÓNIMO	METODOLOGÍA/AUTOR	ACRÓNIMO
Synthesis and Structuring, Reenskaug et al.	OORASS	Desfray, Softeam	Desfray

Tabla 55: Otras Metodologías Orientadas a Objetos

8.4.3 Metodologías de Desarrollo Ágil

XP eXtreme Programming: 1991			
ORGANIZACIÓN	Creada por Kent Beck y Ward Cunningham en 1999, los cuales hacían parte de la comunidad de desarrolladores de <i>SmallTalk</i> .		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		
JUSTIFICACIÓN	Porque fue la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. Además, XP ha conformado un extenso grupo de seguidores en todo el mundo, disparando una gran cantidad de libros que la han convertido en la metodología ágil de más renombre en la actualidad.		
PROPÓSITO	<ul style="list-style-type: none"> ▪ Potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. ▪ Relegar a las metodologías clásicas habitualmente inspiradas en la ingeniería industrial. 		
CONTENIDO	La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.		

XP eXtreme Programming: 1991	
CONTENIDO	<p>Está organizada con base en las siguientes características:</p> <ul style="list-style-type: none">➤ Historias de usuario: Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Dichas historias son descompuestas en tareas de programación (task card) y asignadas a los programadores para ser implementadas durante una iteración.➤ Roles: Los roles de acuerdo con la propuesta original son:<ul style="list-style-type: none">▪ Programador: Escribe las pruebas unitarias y produce el código del sistema.▪ Cliente: Escribe las historias de usuario y las pruebas funcionales para validar su implementación.▪ Encargado de pruebas (Tester): Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.▪ Encargado de seguimiento (Tracker): Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.▪ Entrenador (Coach): Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.▪ Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.▪ Gestor (Big boss): Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.➤ Procesos: El modelo de proceso es eminentemente iterativo, centrando el desarrollo del proyecto en ciclos relativamente cortos de entre 1 y 3 semanas, los cuales consisten (a grandes rasgos) en los siguientes pasos:

XP eXtreme Programming: 1991	
CONTENIDO	<ol style="list-style-type: none">1. El cliente define el valor de negocio a implementar.2. El programador estima el esfuerzo necesario para su implementación.3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.4. El programador construye ese valor de negocio.5. Vuelve al paso 1. <p>En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden.</p> <p>El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.</p> <p>➤ Prácticas: Consiste en la aplicación disciplinada de las siguientes prácticas:</p> <ul style="list-style-type: none">▪ El juego de la planificación: Hay una comunicación frecuente el cliente y los programadores.▪ Entregas pequeñas: Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema.▪ Metáfora: El sistema es definido mediante un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. (Dominio del problema, nomenclatura de clases y métodos del sistema).▪ Diseño simple: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.▪ Pruebas: La producción de código está dirigida por las pruebas unitarias.▪ Refactorización (Refactoring): Consiste en mejorar la estructura interna del código (reestructuración) sin alterar su comportamiento externo, con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

XP eXtreme Programming: 1991	
CONTENIDO	<ul style="list-style-type: none">▪ Programación en parejas: Toda la producción de código debe realizarse con trabajo en parejas de programadores.▪ Propiedad colectiva del código: Cualquier programador puede cambiar cualquier parte del código en cualquier momento.▪ Integración continua: Cada pieza de código es integrada en el sistema una vez que esté lista.▪ 40 horas por semana: Máximo de horas que se debe trabajar por semana.▪ Cliente in-situ: El cliente tiene que estar presente y disponible todo el tiempo para el equipo.▪ Estándares de programación: Enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.
VENTAJAS	<ul style="list-style-type: none">▪ Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.▪ Es adecuada para proyectos con requisitos imprecisos, muy cambiantes y donde existe un alto riesgo técnico.▪ Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.▪ Está preparada para enfrentarse a cambios en los requisitos del proyecto, lo cual es fatídico en todo proyecto software.▪ Técnicas como la integración continua, las pruebas unitarias o el empleo de estándares de codificación son perfectamente compatibles en la mayoría de las metodologías.▪ Consigue evitar problemas habitualmente tardíos que desembocan en retrasos en la planificación inicial del proyecto.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

XP eXtreme Programming: 1991	
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Metodología apropiada solo cuando el tamaño del proyecto es pequeño o medio. ▪ Requiere de equipos de gente calificada y con buena experiencia. ▪ La presencia continua del cliente en el equipo de desarrollo es una premisa difícil de cumplir. ▪ Implica un interés y esfuerzo por parte del cliente que muchas veces no se dará en proyectos reales. ▪ Código como única fuente de documentación, ya que desprecia la documentación en demasía. ▪ El salto de abstracción entre los requisitos y el código es demasiado grande como para que un miembro nuevo en el equipo comprenda el sistema, dada la ausencia de documentación.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega es demasiado limitado. Su modelo de procesos, sus prácticas, la asignación de roles y demás postulaciones que presenta podrían ser de gran ayuda a la hora de determinar las técnicas a implementar en una nueva propuesta.
Tabla 56: Análisis de la Metodología XP - eXtreme Programming (1991)	

SCRUM			
ORGANIZACIÓN	Creado por Ken Schwaber, Jeff Sutherland y Mike Beedle en 1986 y formalizado en 1995		
TIPO	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>
	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SCRUM	
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
JUSTIFICACIÓN	<p>Porque además de ser una de las metodologías ágiles más reconocidas y utilizadas en la gestión de proyectos, se hace muy interesante que sus raíces originales se hayan presentado en el continente asiático, ya que la idea de Ken Schwaber y Jeff Sutherland de formar el proceso hoy conocido como Scrum se produjo a partir de un artículo de la Harvard Business Review titulado "The New New Product Development Game" de Hirotaka Takeuchi e Ikujiro Nonaka, que introducía las mejores prácticas más utilizadas en 10 compañías japonesas altamente innovadoras, lo cual se convierte en un muy buen referente para el mismo.</p>
PROPÓSITO	<ul style="list-style-type: none"> ▪ Definir un marco para la gestión de proyectos. ▪ Elevar al máximo la productividad de un equipo. ▪ Maximizar la realimentación sobre el desarrollo pudiendo corregir problemas y mitigar riesgos de forma temprana. ▪ Reducir al máximo la burocracia y actividades no orientadas a producir software que funcione. ▪ Producir resultados en periodos muy breves de tiempo (cada 30 días), por medio de iteraciones o Sprints.
CONTENIDO	<p>Es un método iterativo e incremental, puesto que el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.</p> <p>Scrum, aplicado al desarrollo de software tiene los siguientes elementos:</p> <ul style="list-style-type: none"> ▪ Roles: Propietario del producto, Gestor o Manager del Scrum, Equipo e Interesados. ▪ Componentes del proceso: Pila del producto (Product Backlog), Pila del sprint (Sprint Backlog), Incremento. ▪ Reuniones: Planificación del sprint, Revisión diaria, Revisión del sprint. <p>Artefactos de Scrum:</p> <ul style="list-style-type: none"> ▪ Sprint: Es la base del desarrollo Scrum. Su duración máxima es de 30 días. Se llevan a cabo las tareas pre-establecidas y no se puede modificar el trabajo acordado en el backlog (registro de acumulación o retraso).

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SCRUM	
CONTENIDO	<ul style="list-style-type: none"> ▪ Product Backlog: Es un documento dinámico que incorpora las constantes necesidades del sistema (requisitos iniciales) y se mantiene durante todo el ciclo de vida (hasta la retirada del Sistema). ▪ Sprint Backlog: Especifica la serie de tareas que se van a desarrollar según los requisitos señalados. <p>El ciclo de vida de Scrum es el siguiente:</p> <ol style="list-style-type: none"> 1. Pre-Juego: Planeamiento. El propósito es establecer la visión, definir expectativas y asegurarse la financiación. 2. Pre-Juego: Montaje (Staging). El propósito es identificar más requerimientos y priorizar las tareas para la primera iteración. Las actividades son planificación, diseño exploratorio y prototipos. 3. Juego o Desarrollo. El propósito es implementar un sistema listo para entrega en una serie de iteraciones de treinta días llamadas “corridas” (sprints). 4. Pos-Juego: Liberación. El propósito es el despliegue operacional.
VENTAJAS	<ul style="list-style-type: none"> ▪ Ideal para proyectos con un rápido cambio de requerimientos. ▪ Enfatiza prácticas y valores de gestión de proyectos por sobre las demás disciplinas del desarrollo. ▪ Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software. ▪ Se puede combinar con otras metodologías de desarrollo ágil como XP para completar sus carencias. ▪ Equipos de entre 6 y 10 personas revisan los requisitos, la tecnología disponible y evalúan los conocimientos para colectivamente determinar como incrementar la funcionalidad. ▪ Se hacen reuniones diarias a lo largo proyecto, antes de empezar a trabajar, con una duración máxima de 4 hrs, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. ▪ Se llevan a cabo hasta que el proyecto este listo para ser puesto en producción o ser lanzado al mercado.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SCRUM	
VENTAJAS	<ul style="list-style-type: none"> ▪ Es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Sólo abarca prácticas de gestión sin entrar en las prácticas de desarrollo como puede hacer XP. ▪ Delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles. ▪ No propone el uso de ninguna práctica de desarrollo en particular.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque haría grandes aportes a una nueva propuesta, teniendo en cuenta que es un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la impredecibilidad y el riesgo a niveles aceptables.
Tabla 57: Análisis de la Metodología Scrum (1995)	

CM Crystal Methodologies							
ORGANIZACIÓN	Creado por Alistair Cockburn en 1998						
TIPO	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Porque es una de las familias de metodologías más conocidas y hace especial énfasis en aspectos tales como la importancia de las personas, el tamaño del equipo de desarrollo y la comunicación, entre otras prioridades.						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

CM Crystal Methodologies			
PROPÓSITO	<p>Proveer un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos.</p>		
CONTENIDO	<p>Crystal da vital importancia a las personas que componen el equipo de un proyecto, y por tanto sus puntos de estudio son:</p> <ul style="list-style-type: none"> ▪ Aspecto humano del equipo ▪ Tamaño de un equipo (número de componentes) ▪ Comunicación entre los componentes ▪ Espacio físico de trabajo ▪ Distintas políticas a seguir <p style="padding-left: 20px;">– Se utilizan políticas diferentes para equipos diferentes. – Codificación por colores dependiendo del tamaño del equipo.</p> <p>Presenta los siguientes roles:</p> <ul style="list-style-type: none"> ▪ Executive Sponsor (Patrocinador Ejecutivo) ▪ Project Manager (Jefe de Proyecto) ▪ Domain Expert (Experto en el Dominio) ▪ Usage Expert (Experto de uso) ▪ Designer-Programmer (Programador Diseñador) ▪ UI Designer (UI Diseñador) ▪ Tester (Realizador de Pruebas) ▪ Technical (Programador Técnico) <p>Utiliza las siguientes herramientas y técnicas:</p> <table style="width: 100%; border: none;"> <tr> <td style="vertical-align: top;"> <p><u>HERRAMIENTAS:</u></p> <ul style="list-style-type: none"> - Sampler Catalog - Use Cases - Non funcional Reqts - Architecture - Tests Cases - UI Design </td> <td style="vertical-align: top; padding-left: 20px;"> <p><u>TÉCNICAS:</u></p> <ul style="list-style-type: none"> - Writing Use Cases - CRC Cards - Responsabiliy Design - Program Derivation </td> </tr> </table> <p>Las metodologías de Crystal más conocidas son las siguientes:</p> <ul style="list-style-type: none"> ▪ Crystal Clear: Se corresponde con el color Blanco en la codificación de colores de Crystal (3 – 8 personas) ▪ Crystal Orange: Se corresponde con el color Naranja en la codificación de colores de Crystal (25 – 50 personas). 	<p><u>HERRAMIENTAS:</u></p> <ul style="list-style-type: none"> - Sampler Catalog - Use Cases - Non funcional Reqts - Architecture - Tests Cases - UI Design 	<p><u>TÉCNICAS:</u></p> <ul style="list-style-type: none"> - Writing Use Cases - CRC Cards - Responsabiliy Design - Program Derivation
<p><u>HERRAMIENTAS:</u></p> <ul style="list-style-type: none"> - Sampler Catalog - Use Cases - Non funcional Reqts - Architecture - Tests Cases - UI Design 	<p><u>TÉCNICAS:</u></p> <ul style="list-style-type: none"> - Writing Use Cases - CRC Cards - Responsabiliy Design - Program Derivation 		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

CM Crystal Methodologies	
VENTAJAS	<ul style="list-style-type: none"> ▪ El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado únicamente por los recursos a utilizar. ▪ El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. ▪ Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). ▪ La mejora individual constituye una mejora global del equipo. ▪ Cada metodología tiene unas prioridades a la hora de intentar alcanzar el éxito de la aplicación. ▪ La familia de Crystal busca una combinación de productividad y tolerancia. ▪ La comunicación es más barata y mejor cuanto más “cercana” sea. ▪ Crystal recomienda la interacción cara a cara, por ser éste el mejor método de comunicación.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Cuantas más personas estén implicadas, más grande debe ser la metodología. ▪ Si el proyecto tiene mucha densidad, un error no detectado puede ser crítico. ▪ El aumento de tamaño o densidad añade un coste considerable al proyecto.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque para éstas metodologías el éxito del proyecto depende casi que exclusivamente de las personas que componen el equipo, y no consideran muchos otros aspectos igualmente importantes.
Tabla 58: Análisis de la Familia Metodológica Crystal (1998)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

DSDM Dynamic Systems Development Method							
ORGANIZACIÓN	Creado por un consorcio formado originalmente por 17 compañías del Reino Unido (miembros fundadores) en Enero de 1994						
TIPO	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	Porque es una metodología muy flexible que puede ser combinada con otras metodologías de desarrollo ágil como XP (conocida como EnterpriseXP) o incluso con <i>metodologías pesadas</i> como RUP de Rational, en la que se demuestra su la compatibilidad a pesar de sus fuertes diferencias terminológicas.						
PROPÓSITO	<ul style="list-style-type: none"> ▪ Definir el marco para desarrollar un proceso de producción de software. ▪ Crear una metodología RAD unificada. 						
CONTENIDO	Su estructura se guía por los siguientes principios: <ol style="list-style-type: none"> 1. El involucramiento del usuario es imperativo. 2. Los equipos de DSDM deben tener el poder de tomar decisiones. 3. El foco está puesto en la entrega frecuente de productos. 4. La conformidad con los propósitos del negocio es el criterio esencial para la aceptación de los entregables. 5. El desarrollo iterativo e incremental es necesario para converger hacia una correcta solución del negocio. 6. Todos los cambios durante el desarrollo son reversibles. 7. Los requerimientos están especificados a un alto nivel. 8. El testing es integrado a través del ciclo de vida. 9. Un enfoque colaborativo y cooperativo entre todos los interesados es esencial. 						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

DSDM Dynamic Systems Development Method	
CONTENIDO	Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.
VENTAJAS	<ul style="list-style-type: none"> ▪ Es un proceso iterativo e incremental. ▪ El equipo de desarrollo y el usuario trabajan juntos. ▪ El equipo mínimo de DSDM es de dos personas y puede llegar a seis, pero puede haber varios equipos en un proyecto. ▪ El mínimo de dos personas involucra que un equipo consiste de un programador y un usuario. ▪ Es aplicable a proyectos grandes y pequeños. ▪ Es considerada la primera metodología ágil y la que más se aproxima a los métodos tradicionales. ▪ Su implantación incluso permitiría alcanzar un nivel 2 de madurez según CMMI.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Para ser aplicado en sistemas grandes es necesaria su partición en componentes para ser desarrollados por equipos normales. ▪ No es muy utilizada fuera del Reino Unido.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>
¿POR QUÉ?	Porque representa una alternativa de gran trayectoria y flexibilidad a la hora de desarrollar procesos de producción de software. Además, por ser concebida como una metodología de desarrollo ágil con el objetivo de unificar otras metodologías, reúne en gran parte los planteamientos utilizados por las mismas, lo cual podría representar un gran aporte a una nueva propuesta para Colombia.
Tabla 59: Análisis de la Metodología DSDM (1994)	

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

FDD Feature Driven Development							
ORGANIZACIÓN	Creado por Peter Coad, Eric Lefebvre y Jeff De Luca en 1998 y retomada por Palmer & Felsing en el 2002						
TIPO	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Estándar <input type="checkbox"/></td> <td style="width: 33%;">Norma <input type="checkbox"/></td> <td style="width: 33%;">Guía <input type="checkbox"/></td> </tr> <tr> <td>Modelo <input type="checkbox"/></td> <td>Metodología <input checked="" type="checkbox"/></td> <td></td> </tr> </table>	Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>	Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>	
Estándar <input type="checkbox"/>	Norma <input type="checkbox"/>	Guía <input type="checkbox"/>					
Modelo <input type="checkbox"/>	Metodología <input checked="" type="checkbox"/>						
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>						
JUSTIFICACIÓN	<p>Porque a diferencia de otras metodologías ágiles no cubre todo el ciclo de vida sino sólo las fases de diseño y construcción y se considera adecuado para proyectos mayores y de misión crítica.</p> <p>Además, porque introduce conceptos como “<i>Programación orientada por rasgos (FOP) o características (features)</i>” y “<i>Desarrollo guiado por rasgos (FDD)</i>”, que aunque tienen coincidencias entre si, no necesariamente deben ser implementadas conjuntamente.</p>						
PROPÓSITO	<p>Permitir el desarrollo de proyectos software con tiempo de desarrollo relativamente cortos, a través de las características o funcionalidades significativas para el cliente.</p> <p>Definir métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar a conocer el estado actual del desarrollo y a realizar mejores estimaciones en proyectos futuros.</p>						
CONTENIDO	<p>Se basa en un proceso iterativo con iteraciones cortas (hasta 2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear.</p> <p>Las iteraciones se deciden en base a <i>features</i> (características) o funcionalidades, que son pequeñas partes del software con significado para el cliente.</p> <p>Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.</p> <p>FDD divide un proyecto en 5 fases:</p> <ol style="list-style-type: none"> 1. Desarrollo de un modelo general 2. Construcción de la lista de funcionalidades 						

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

FDD Feature Driven Development	
CONTENIDO	<p>3. Plan de <i>releases</i> (versiones) en base a las funcionalidades a implementar</p> <p>4. Diseñar en base a las funcionalidades</p> <p>5. Implementar en base a las funcionalidades</p> <p>Roles en FDD:</p> <ul style="list-style-type: none"> ➤ Roles claves <ul style="list-style-type: none"> ▪ Administrador del proyecto ▪ Arquitecto jefe ▪ Administrador de desarrollo ▪ Programador jefe ▪ Propietarios de clases (programadores) ▪ Experto de dominio (cliente, patrocinador, analista de negocios o una mezcla de todo eso) ➤ Roles de soporte <ul style="list-style-type: none"> ▪ Administrador de entrega ▪ Abogado/guru de lenguaje ▪ Ingeniero de construcción ▪ Herramientista (toolsmith) ▪ Administrador del sistema ➤ Roles adicionales <ul style="list-style-type: none"> ▪ Verificadores ▪ Encargados del despliegue <p>Escritores técnicos</p>
VENTAJAS	<ul style="list-style-type: none"> ▪ Es un método ágil, iterativo y adaptativo que permite entregar al cliente software funcional en poco tiempo. ▪ Es adecuado para proyectos mayores y de misión crítica. ▪ No requiere un modelo específico de proceso y se complementa con otras metodologías. ▪ Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso. ▪ Sus principios son pocos y simples. ▪ No exige la presencia del cliente.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ No cubre todo el ciclo de vida sino sólo las fases de diseño y construcción. ▪ Debido a sus múltiples roles, hay quienes dicen que es demasiado jerárquico para ser un método ágil. ▪ El código fuente tiene propietario.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

FDD Feature Driven Development	
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Documentación aceptable. ▪ Es necesario tener en el equipo miembros con gran experiencia. ▪ Ausencia de procedimientos detallados de prueba. ▪ Es necesario implementarlo de manera gradual en proyectos nuevos o actualizaciones de sistemas existentes.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	<p>Porque se podría considerar a medio camino entre RUP y XP, aunque más similar a éste último por seguir siendo un proceso ligero. Además, muchas de sus características son similares a las planteadas en dichas metodologías, aunque con un enfoque diferente.</p> <p>También porque no cubre todo el ciclo de vida sino sólo las fases de diseño y construcción.</p> <p>Por otro lado, FDD es marca registrada de la empresa Nebulon Pty de Australia.</p>
Tabla 60: Análisis de la Metodología DFF (1998 y 2002)	

MSF Versión 4.0 Microsoft Solution Framework			
ORGANIZACIÓN	Creado por Microsoft Corporation en el 2005		
TIPO	Estándar <input type="checkbox"/> Modelo <input type="checkbox"/>	Norma <input type="checkbox"/> Metodología <input checked="" type="checkbox"/>	Guía <input checked="" type="checkbox"/>
ANALIZADA EN LA INVESTIGACIÓN	Si <input checked="" type="checkbox"/> No <input type="checkbox"/>		

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

MSF Versión 4.0 Microsoft Solution Framework	
JUSTIFICACIÓN	<p>Porque es una metodología para el desarrollo de software que permite la planificación, desarrollo y gestión de proyectos tecnológicos creada por una de las compañías multinacionales de mayor éxito en el desarrollo, fabricación, licenciamiento y producción de software y equipos electrónicos para el mercado informático desde su creación en 1975.</p>
PROPÓSITO	<ul style="list-style-type: none"> ▪ Presentar de manera flexible e interrelacionada una serie de conceptos, modelos y mejores prácticas de uso que sirvan de guía para controlar la planificación, el desarrollo y la gestión de proyectos tecnológicos. ▪ Servir como metodología para el desarrollo de software centrada en los modelos de proceso y de equipo, dejando en un segundo plano las elecciones tecnológicas.
CONTENIDO	<p>Se compone de principios, modelos y disciplinas.</p> <p>Principios:</p> <ol style="list-style-type: none"> 1. Promover comunicaciones abiertas. 2. Trabajar para una visión compartida. 3. Fortalecer los miembros del equipo. 4. Establecer responsabilidades claras y compartidas. 5. Focalizarse en agregar valor al negocio. 6. Permanecer ágil y esperar los cambios. 7. Invertir en calidad. 8. Aprender de todas las experiencias. <p>Disciplinas:</p> <ul style="list-style-type: none"> ➤ Gestión de Proyectos: Permite mayor escalabilidad, desde proyectos pequeños a proyectos largos y complejos. Se basa en: <ul style="list-style-type: none"> ▪ Planificar sobre entregas cortas. ▪ Incorporar nuevas características sucesivamente. ▪ Identificar cambios ajustando el cronograma. ➤ Control de Riesgos: Identificar las prioridades, tomar decisiones y controlar las emergencias. ➤ Control de Cambios: Registrar y hacer evidentes los cambios. <p>Modelos: Se aplican a diferentes categorías de problemas que están enmarcadas en la planeación, construcción o implantación, y administración de los sistemas. MSF se centra en los modelos de proceso y de equipo.</p>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
 Universidad Católica Popular del Risaralda

MSF Versión 4.0 Microsoft Solution Framework Agile	
CONTENIDO	<ul style="list-style-type: none"> ➤ Proceso (Process Model): Es una combinación del modelo en cascada y el de espiral. Permite mejorar el control del proyecto, minimizando el riesgo y aumentar la calidad acortando el tiempo de entrega. Está compuesto por las siguientes fases: <ul style="list-style-type: none"> ▪ Visión: Requerimientos y objetivos ▪ Planeación: Plan maestro y cronograma. ▪ Desarrollo: Versiones entregables del proyecto. ▪ Estabilización: Pruebas del producto. ▪ Implantación: Puesta en funcionamiento. ▪ Soporte: Garantía del producto. ➤ Equipo de trabajo (Team Model): Alienta la agilidad para hacer frente a nuevos cambios involucrando a todo el equipo en las decisiones fundamentales. ➤ Arquitectura Empresarial (Enterprise Architecture Model) ➤ Aplicaciones (Application Model) ➤ Diseño de Soluciones con Componentes (Designing Component Solutions - DCS)
VENTAJAS	<ul style="list-style-type: none"> ▪ Mantiene siempre el enfoque al usuario. ▪ Reduce el tiempo de desarrollo mediante la reutilización de múltiples componentes. ▪ Equipo de trabajo balanceado, con tareas y objetivos claramente definidos. ▪ Permite saber en todo momento cuál es el grado real de avance del proyecto. ▪ Permite determinar cuáles son los riesgos que se corren si se decide introducir modificaciones al proyecto una vez que el desarrollo ha sido iniciado. ▪ Es más ágil que sus predecesores (MSF 1.0 de 1993 y MSF 3.0 de 2002) ▪ Cuenta con el soporte y mantenimiento de Microsoft. ▪ Aplicable a grandes y pequeños proyectos.
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Su aplicación constituye un trabajo bastante largo, ya que cada fase se debe documentar profundamente. ▪ Por ser un producto de Microsoft, trata de imponer sus herramientas a la hora de implementar MSF.

MSF Versión 4.0 Microsoft Solution Framework Agile	
DESVENTAJAS	<ul style="list-style-type: none"> ▪ Los precios de las licencias, capacitación y soporte por parte de Microsoft son caros. ▪ Requiere de mucha documentación.
¿DEBERÍA SER INCLUIDA EN UNA NUEVA PROPUESTA?	Si <input type="checkbox"/> No <input checked="" type="checkbox"/>
¿POR QUÉ?	Porque a pesar de ser una metodología flexible que se puede combinar con otras complementarias, su uso está muy ligado al entorno Microsoft y existen otras metodologías que igualmente permiten alcanzar los mismos objetivos. Además, se centra mucho en los modelos de proceso y de equipo descuidando un poco otros aspectos igualmente importantes.
Tabla 61: Análisis de la Metodología MSF Versión 4.0 (2005)	

8.5 Propuestas para el aseguramiento de la calidad del software en Colombia

Según el libro *“Desafíos y Oportunidades de la Industria del Software en América Latina”* el cual es resultado de investigaciones realizadas en seis países – Argentina, Brasil, Chile, Colombia, México y Uruguay – como parte de una iniciativa más amplia de la Comisión Económica para América Latina y el Caribe (CEPAL¹¹) en analizar diferentes aspectos económicos y sociales de las tecnologías de la información y de la comunicación, y buscando evaluar el potencial, las dificultades y las estrategias de los países latinoamericanos para participar de la industria global de software y

¹¹ <http://www.eclac.org>. La CEPAL es una de las cinco comisiones regionales de las Naciones Unidas y su sede está en Santiago de Chile. Se fundó para contribuir al desarrollo económico de América Latina, coordinar las acciones encaminadas a su promoción y reforzar las relaciones económicas de los países entre sí y con las demás naciones del mundo. Posteriormente, su labor se amplió a los países del Caribe y se incorporó el objetivo de promover el desarrollo social.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

servicios de informática (SSI); la industria de software en Colombia se encuentra bastante desarticulada. Falta camino por recorrer, aun cuando se está trabajando para el fortalecimiento de la agremiación de las empresas de software. La desarticulación no sólo está presente entre las empresas locales sino entre el Estado y las federaciones de software¹² y entre éstas y las empresas. (CEPAL, 2009)

La falta de sincronía, de acción conjunta y, especialmente, de comunicación son las debilidades más grandes que tiene esta industria en el país, pues hacen que el sector no sea explotado de acuerdo con su potencial. En este mismo sentido, es sorprendente la falta de consolidación de datos tanto por parte del gobierno como de las federaciones. Siendo un sector tan dinámico, de alta tecnología y manejo de bases de datos, es irónico que la actualización de cifras e información no se haga de manera periódica y que incluso, en ocasiones, difieran sustancialmente las cifras de las diferentes fuentes. Esta falta de comunicación tiene, por lo tanto, repercusiones directas sobre el desarrollo de la industria de software como unidad y como estrategia nacional. CEPAL et al. (2009, cap. 5, pág. 139) Dado lo anterior, se puede decir que en la actualidad no se cuentan con los elementos suficientes para aventurarse a reafirmar o derrumbar de forma acertada la hipótesis de que el software producido en Colombia es de baja calidad.

Por otro lado, es importante destacar que la industria de software de Colombia está dominada por microempresas y pequeñas empresas, que se dedican especialmente al desarrollo de software a medida, a la intermediación entre las multinacionales y los

¹² En Colombia existen principalmente dos federaciones: una es la Federación Colombiana de la Industria de Software (Fedesoft) <http://fedesoft.org> que representa principalmente a las pequeñas empresas locales de software, y la segunda es Business Software Alliance (BSA) <http://www.bsa.org> una organización internacional sin ánimo de lucro creada para promover los objetivos de la industria del software y sus socios del hardware de la cual Colombia es miembro.

clientes finales, a la compra venta de equipos y a la oferta de diferentes servicios relacionados con el sector de TI. Según lo expresado en el estudio **“Colombia: Desafíos de una Industria en Formación¹³”**, hasta el año 2009 habían 130 empresas con certificación ISO 9000, una empresa con certificación CMM 5 y otras cinco estaban en proceso de obtener esta certificación, lo que ayuda a reafirmar que el país no cuenta con una propuesta metodológica para estandarizar el proceso de construcción y evaluación del producto software que permita medir la calidad del mismo.

De igual forma, se menciona que las empresas netamente colombianas son reconocidas como “artesanas del software” pues se han concentrado en el desarrollo de software a medida, es decir, adecuado especialmente a las necesidades del cliente, lo que da a entender que en muchas ocasiones no se utiliza como referente un estándar o metodología de desarrollo existente en el ámbito internacional.

A partir de la situación expuesta anteriormente, en Colombia han surgido algunas iniciativas que propenden por el fortalecimiento de la industria del software en el país, entre ellas se encuentran: El Programa de Apoyo a las Tecnologías de la Información (PRATI) y el Sistema Integral de Mejoramiento de los Procesos de Desarrollo de Software (SIMEP-SW). De igual forma, la nación participó en el proyecto de cooperación internacional para la Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica denominado COMPETISOFT.

¹³ Capítulo 5 del Libro *“Desafíos y Oportunidades de la Industria del Software en América Latina”* publicado por la CEPAL en el año 2.009.

8.5.1 PRATI

El Ministerio de Comunicaciones presentó en el año 2008 el Programa de Apoyo a las Tecnologías de Información (PRATI) como la estrategia propuesta para hacer del software un sector de talla mundial mediante un árbol de objetivos e iniciativas, la cual cubría las diferentes categorías empleadas para medir el índice de competitividad de la industria de las TI y de las firmas especializadas en localización de inversión. Dentro de sus indicadores principales se encontraba: La asociatividad y consolidación de la industria, el crecimiento e inversión en TI, el acceso a créditos y capital de riesgo, el estudio de la calidad del talento humano, el estado del entorno (Parque tecnológicos funcionando, número de empresas con certificaciones internacionales, infraestructura y tasa de piratería) y la investigación y desarrollo. (Ministerio de Comunicaciones de Colombia, 2009)

8.5.2 COMPETISOFT

Es un proyecto de Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica, financiado por CYTED¹⁴ que se desarrolló durante el período 2006-2008, con la participación de la Universidad EAFIT, la Universidad del Cauca y Parquesoft Popayán por parte de

¹⁴ http://www.cytmed.org/cytmed_informacion/es/presentacion.php. Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo creado en 1984 mediante un Acuerdo Marco Interinstitucional firmado por 19 países de América Latina, España y Portugal. Se define como un programa intergubernamental de cooperación multilateral en Ciencia y Tecnología, que contempla diferentes perspectivas y visiones para fomentar la cooperación en Investigación e Innovación para el Desarrollo de la Región Iberoamericana. Su objetivo principal es contribuir al desarrollo armónico de la Región Iberoamericana mediante el establecimiento de mecanismos de cooperación entre grupos de investigación de las Universidades, Centros de I+D y Empresas innovadoras de los países iberoamericanos, que pretenden la consecución de resultados científicos y tecnológicos transferibles a los sistemas productivos y a las políticas sociales.

Colombia. Otros países que contribuyeron con su realización fueron: Argentina, Brasil, Chile, Costa Rica, Cuba, Ecuador, El Salvador, España, Guatemala, México, Perú, Portugal, Uruguay y Venezuela. EL principal objetivo de COMPETISOFT fue el incrementar el nivel de competitividad de las PyMES Iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, pudiera llegar a ser la base sobre la cual establecer un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica.

El modelo propuesto se enfocó en procesos y consideraba los tres niveles básicos de la estructura de una organización: Alta Dirección, Gestión y Operación. El modelo pretendía apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. (CYTED, 2010)

8.5.3 SIMEP-SW

El Sistema Integral de Mejoramiento de los Procesos de desarrollo de Software (SIMEP-SW) en Colombia es un proyecto conjunto entre la Universidad de Chile y la Universidad del Cauca, financiado por ésta última, Colciencias¹⁵ y SITIS Ltda.¹⁶, el cual intenta alivianar los requisitos, ser guía en el proceso de mejora y generar un

¹⁵ <http://www.colciencias.gov.co>. Colciencias es el Departamento Administrativo de Ciencia, Tecnología e Innovación, el cual promueve las políticas públicas para fomentar la CTI en Colombia.

¹⁶ <http://www.sitis.info/index.php>. Soluciones Informáticas Integrales Ltda. - SITIS LTDA. es una empresa colombiana, de carácter privado y de base tecnológica dedicada a la investigación, diseño y desarrollo de soluciones en tecnologías blandas y sistemas para empresas e instituciones públicas y privadas, nacionales y extranjeras, especialmente el Ministerio de la Protección Social, Direcciones Territoriales de Salud, Direcciones Locales de Salud, Administradoras del Régimen Subsidiado, Empresas Promotoras de Salud e Instituciones Prestadoras de Servicios de Salud.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

conjunto de recomendaciones prácticas para la implementación de los requisitos del proceso software en las micro, pequeñas y medianas empresas de software. Dicho proyecto surgió con el ánimo de complementar los esfuerzos realizados por algunos países latinoamericanos como Brasil con MPS-Br (Mejoramiento del Proceso de Software) y México con MoProSoft (Modelo de Procesos para la Industria de Software en México), puesto que los mismos buscan el fortalecimiento de la industria de software de cada país, trasladando los requisitos que imponen algunos modelos de reconocimiento internacional tales como CMMI e ISO, pero no presentan esfuerzos orientados a la implementación de estos requisitos dentro o fuera del marco de un proyecto de mejora en el tipo de empresas mencionadas.

El modelo de referencia para SIMEP-SW, según Hurtado & Bastarrica, corresponde al resultado de evaluar los modelos de calidad existentes y al estudio de las prácticas que siguen un conjunto de empresas de desarrollo de software del sur occidente colombiano. Este modelo de calidad debe tener los elementos comunes que le permitan a la organización ir adecuando el proceso para obtener con facilidad una certificación internacional. (Hurtado & Bastarrica, 2005, pág. 6)

Uno de los principales objetivos del proyecto es generar el Modelo de Mejoramiento a seguir por parte de las empresas para sacar adelante procesos de mejoramiento y/o certificación, para lo cual definirá la forma de llevar a cabo un proyecto de mejoramiento dentro de una organización PyME e implementará los elementos de gestión (un proceso, métodos, prácticas y técnicas) necesarios para asegurar su éxito. De igual manera el proyecto busca brindar unas recomendaciones sobre las prácticas, técnicas y métodos a utilizar dentro del proceso de desarrollo de software para las PyMEs.

En el marco del proyecto y en conjunto con el proyecto [UC Activos de procesos], se ha gestado la posibilidad de crear Agile SPsL, una infraestructura conceptual y tecnológica para la implementación de procesos de software ágiles que cumplan con modelos y/o estándares de calidad asociados a los procesos desarrollo de software. (Hurtado & Bastarrica, 2005, pág. 7)

8.6 Herramientas

8.6.1 CASE.

Según Weitzenfeld, existen *herramientas* que apoyan los diversos aspectos del proceso de software. Al conjunto de herramientas aplicables al desarrollo de sistemas de software se les conoce como **CASE (“Computer-Aided Software Engineering”)**, herramientas para asistir al desarrollador en las diferentes fases del ciclo de vida del proceso del software: planeación, requisitos, análisis, diseño, implementación, pruebas (verificación y validación), documentación, mantenimiento y administración. Las herramientas varían en el tipo de componentes que incorporan, editores (textuales y gráficos), programadores (codificadores, depuradores, compiladores y ensambladores), verificadores y validadores (analizadores estáticos y dinámicos y diagramas de flujos), medidores (monitores), administradores de la configuración (versiones y librerías) y administradores del proyecto (estimación, planeación y costo). La herramienta particular a ser usada debe apoyar el modelo de proceso escogido, y no se debe considerar métodos independientes de herramientas. Si las herramientas son buenas, éstas deben resultar en mejoras notorias en la producción del sistema. Las herramientas deben manejar aspectos de administración de información, generar los distintos tipos de diagramas e incluso el código final. La sofisticación de las herramientas varía desde aquellas que apoyan a un sólo desarrollador en un sólo proyecto hasta aquellas que apoyan múltiples

desarrolladores trabajando juntos en un proyecto con almacenamiento compartidos, e incluso múltiples proyectos (Weitzenfeld, 2002, cap. 3, pág. 25). Existen actualmente productos que manejan múltiples métodos y notaciones, tales como UML.

8.6.2 UML (Unified Modeling language).

Es un lenguaje usado para especificar, visualizar y documentar los diferentes aspectos relativos a un sistema de software bajo desarrollo, así como para modelado de negocios y almacenamiento de datos.

Los principales factores que motivaron la creación de UML fueron:

- La necesidad de modelar sistemas.
- Las tendencias en la industria de software.
- Unificar los distintos lenguajes y métodos existentes.
- Innovar los modelos para adaptarse a la arquitectura distribuida.

Reseña histórica

UML comenzó como una iniciativa de Grady Booch y Jim Rumbaugh en 1994 para combinar las notaciones visuales de sus conceptos (Booch y OMT). En 1995 Ivar Jacobson, creador del OOSE se unió al proyecto. En 1997 la versión 1.0 de UML fue propuesta como metamodelo orientado a objetos de semántica y notación de estándares. En 1997 fue presentado como estándar por el OMG¹⁷ (Object

¹⁷ El consorcio OMG está formado por las siguientes compañías: Rational Software, Oracle, IBM, DEC, Microsoft, Hewlett-Packard, Sterling Software, MCI Systemhouse, Unisys, IntelliCorp, ICON Computing, i-Logix, ObjectTime, Platinum Technology, Petch, Taskon A/S, Reich Technologies, Softeam. Universidad de Huelva. (s.f.). *El lenguaje unificado de modelado, UML*. Recuperado el 21 de Mayo de 2010, de Universidad de Huelva: http://www.uhu.es/juanmanuel.cordoba/old/d_is2/teoria/tema1/t1a.pdf

Management Group). Para el 99 se hablaba de la versión 1.2 y para el 2002 de UML 1.4 y 2.0, la cual es la última versión aprobada.

Características

- Es una especificación basada en Booch, OMT y OOSE, de allí sus principios.
- Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
- Permite describir un sistema en diferentes niveles de abstracción.
- Se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del desarrollo de una aplicación, sin definir un modelo de desarrollo.

Modelos

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- **Diagramas de Casos de Uso:** Modelado de los procesos.
- **Diagramas de Secuencia:** Modelado del paso de mensajes entre objetos.
- **Diagramas de Colaboración:** Modelado de las interacciones entre objetos.
- **Diagramas de Estado:** Modelado del comportamiento de los objetos en el sistema.
- **Diagramas de Actividad:** Modelado del comportamiento de los Casos de Uso, objetos y operaciones.
- **Diagramas de Clases:** Modelado de la estructura estática de las clases en el sistema.
- **Diagramas de Objetos:** Modelado de la estructura estática de los objetos en el sistema.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E
INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

- **Diagramas de Componentes:** Modelado de componentes.
- **Diagramas de Implementación:** Modelado de la distribución del sistema.

9. CONCLUSIONES Y RECOMENDACIONES

- La dinámica de la ingeniería del software como disciplina ha llevado a organizaciones de diferentes países a proponer modelos/metodologías/guías para el aseguramiento de la calidad del software; lo que deja entrever que, por su complejidad, el software no podría ser tratado con un solo modelo estándar a nivel mundial, sino que por el contrario, cada nación, y en algunos casos, cada región, debe formular una propuesta que le permita fabricar software confiable para el usuario final y satisfactorio para su(s) creador(es). La propuesta resultante no debe desconocer los avances expresados en este trabajo como estado del arte, sino que deben ser el motor que impulse la implementación de la misma.
- La aplicación de buenas prácticas de ingeniería, es de vital importancia en la construcción de productos software de buena calidad. De la información explorada se deduce que las propuestas que mayor vigencia muestran, son las que lograron tratar con mayor seriedad la disciplina, como rama de la ingeniería.
- Existe una cantidad considerable de organizaciones que agremian a la industria del software a nivel mundial, y son éstas quienes en su gran mayoría proponen los estándares a seguir para desarrollar software de calidad.
- En la actualidad, existe un gran número de Estándares, Normas, Guías, Modelos, Metodologías y Métodos que se pueden aplicar en la producción, evaluación y mantenimiento de software. Sin embargo, muchos de ellos siguen siendo poco conocidos y/o implementados a pesar de los buenos resultados obtenidos gracias a su aplicación.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

- No todas las prácticas de ingeniería analizadas en la investigación, son aplicables a las organizaciones al momento de planear, gestionar, producir, evaluar y mantener software de calidad. Por lo anterior, es importante que, en la formulación de una nueva propuesta de aseguramiento de la calidad se evalúe la aplicación de las existentes con el fin de lograr criterio a la hora de decidir su inclusión o no como referencia.
- En la exploración nacional no se evidencia modelo, metodología o guía que permita a los fabricantes de software aplicar un estándar de manera sistematizada, lo que obstaculiza a las organizaciones el cumplimiento de normas de calidad en el proceso y por consiguiente en el producto; se deduce que lo anterior ha sido factor clave de falta de competitividad en el mercado internacional.
- En Colombia se hace necesario fomentar cada vez más una cultura investigativa, que permita a sus desarrolladores conocer las tendencias existentes en ingeniería del software y a su vez faciliten la generación de productos de calidad.
- Para la elaboración de una propuesta marco para Colombia, se recomienda tener en cuenta las experiencias adquiridas en países de América Latina como México y Brasil que, según muestra el estudio, son los que llevan más tiempo trabajando alrededor del tema.
- Es recomendable que a la hora de realizar una propuesta marco para Colombia, se tengan en cuenta las necesidades propias del entorno, sin desconocer las carencias existentes en el ámbito internacional.

10. BIBLIOGRAFÍA

CEPAL. (2009). *Desafíos y oportunidades de la industria del software en América Latina*. (P. Bastos Tigre, & F. Silveira Marquez, Edits.) Colombia: Mayol Ediciones S.A.

Peláez Valencia, L. E. (2009). *METRICA V.3 - Presentación*. Pereira.

Piattini Velthuis, M. G. (1996). *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Madrid: Rama.

Pressman, R. S. (2002). El Modelo de Construcción de Prototipos. En R. S. Pressman, *Ingeniería del Software un Enfoque Práctico Quinta Edición* (págs. 21-22). Madrid: Mc Graw Hill.

Pressman, R. S. (2002). El Modelo DRA. En R. S. Pressman, *Ingeniería del Software Un Enfoque Práctico Quinta Edición* (págs. 22-23). Madrid: Mc Graw Hill.

Pressman, R. S. (2002). El Modelo Espiral. En R. S. Pressman, *Ingeniería del Software un Enfoque Práctico Quinta Edición* (págs. 24-26). Madrid: Mc Graw Hill.

Rumbaugh, J., Blaha, M., Lorensen, W., & Frederick, E. (1991). *Object-Oriented Modeling and Design*. Prentice-Hall.

Sommerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson Education S.A.

Weitzenfeld, A. (2002). *INGENIERIA DE SOFTWARE ORIENTADA A OBJETOS*. México: Thomson.

11. WEBGRAFÍA

AESOFT. (s.f.). *Estatutos de la Asociación "AESOFT"*. Recuperado el 05 de Diciembre de 2009, de AESOFT:

http://www.aesoft.com.ec/index.php?option=com_content&task=view&id=114&Itemid=69

AESOFT. (2009). *Qué es la AESOFT*. Recuperado el 05 de Diciembre de 2009, de Asociación Ecuatoriana de Software:

http://www.aesoft.com.ec/index.php?option=com_content&task=view&id=108&Itemid=56

AMITI. (2010). *AMITI*. Recuperado el 16 de Enero de 2010, de Asociación Mexicana de la Industria de Tecnologías de Información, A.C.: <http://www.amiti.org.mx>

Angeleri, P. M. (s.f.). *Polo IT Buenos Aires*. Recuperado el 19 de Febrero de 2010, de <http://www.itbuenosaires.com.ar>:

<http://www.itbuenosaires.com.ar/Prensa/Art%C3%ADculo%20IRAM.pdf>

ANSI. (s.f.). *ISO 9241-11:1998*. Recuperado el 05 de Marzo de 2010, de American National Standards Institute:

<http://webstore.ansi.org/RecordDetail.aspx?sku=ISO+9241-11%3a1998&source=google&adgroup=iso8&keyword=iso%209241-11&gclid=CLyx-pXWoKACFQxinAodMmMTaA>

ANSI. (s.f.). *ISO 9241-12:1998*. Recuperado el 05 de Marzo de 2010, de American National Standards Institute:

http://webstore.ansi.org/RecordDetail.aspx?sku=ISO+9241-12%3a1998&source=google&adgroup=iso8&keyword=iso%209241-12&gclid=CITKpIXYoKACFQ-bnAodyE8_Zw

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ANSI. (s.f.). *ISO 9241-13:1998*. Recuperado el 06 de Marzo de 2010, de American National Standards Institute :

http://webstore.ansi.org/RecordDetail.aspx?sku=ISO+9241-13%3a1998&source=google&adgroup=iso8&keyword=iso%209241-13&gclid=CLaDqd6Mo6ACFRJinAod_hbZZQ

ANSI. (s.f.). *ISO 9241-4:1998*. Recuperado el 27 de Febrero de 2010, de American National Standards Institute:

<http://webstore.ansi.org/RecordDetail.aspx?sku=BS+EN+ISO+9241-4%3a1998>

ANSI. (s.f.). *ISO 9241-6:1999*. Recuperado el 05 de Marzo de 2010, de American National Standards Institute :

http://webstore.ansi.org/RecordDetail.aspx?sku=ISO+9241-6%3a1999&source=google&adgroup=iso8&keyword=iso%209241-6&gclid=CNz9q9_EoKACFRkhnAodmlkWbQ

ANSI. (s.f.). *ISO 9241-7:1998*. Recuperado el 05 de Marzo de 2010, de American National Standards Institute:

<http://webstore.ansi.org/RecordDetail.aspx?sku=BS+EN+ISO+9241-7%3a1998>

ANSI. (s.f.). *ISO 9241-8:1998*. Recuperado el 05 de Marzo de 2010, de American National Standards Institute :

<http://webstore.ansi.org/RecordDetail.aspx?sku=BS+EN+ISO+9241-8%3a1998>

ANSI. (s.f.). *ISO 9241-9:2000*. Recuperado el 05 de Marzo de 2010, de American National Standards Institute:

<http://webstore.ansi.org/RecordDetail.aspx?sku=BS+EN+ISO+9241-9%3a2000>

APESOFTE. (2009). *APESOFTE*. Recuperado el 19 de Diciembre de 2009, de Asociación Peruana de Productores de Software: <http://www.apesoft.org/quienes.htm>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

APESOF. (s.f.). *Proyectos APESOF*. Recuperado el 19 de Diciembre de 2009, de Asociación Peruana de Productores de Software:
<http://www.apesoft.org/proyectos.htm>

APM Group - International. (2010). *APMG - International*. Recuperado el 07 de Abril de 2010, de APMG-UK: <http://www.apmg-international.com/APMG-UK/PRINCE2/PRINCE2Home.asp>

Association for Computing Machinery. (s.f.). *ACM*. Recuperado el 28 de Abril de 2010, de The Association for Computing Machinery, Inc.: <http://www.acm.org/>

Axentia. (2006). *Una Introducción a CMMI*. Recuperado el 21 de Mayo de 2010, de Axentia:
<http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf>

Beltré Ferreras, H. J. (2008). *Aplicación de la Usabilidad al Proceso de Desarrollo de Páginas Web*. Recuperado el 24 de Marzo de 2010, de Archivo Digital - Universidad Politecnica de Madrid:
http://oa.upm.es/1176/1/HAYSER_JACQUELIN_BELTRE_FERRERAS.pdf

BSI. (s.f.). *British Standards Institution*. Recuperado el 20 de Febrero de 2010, de British Standards Institution: <http://www.bsigroup.com/en/>

Caballero Muñoz, I. (Mayo de 1999). *La norma IEEE 1058.1: Plan para la Gestión de Proyectos Software*. Recuperado el 01 de Abril de 2010, de Google code: [pgsi-g5.googlecode.com/files/T9899_ICaballero.pdf](https://code.google.com/files/T9899_ICaballero.pdf)

Calderón, A., Valverde Rebaza, S. D., & J. C. (2007). *Metodologías Ágiles*. Recuperado el 18 de Abril de 2010, de Sociedad de Estudiantes de Ciencia de la Computación: <http://www.seccperu.org/files/Metodologias%20Agiles.pdf>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

CAMTIC. (s.f.). *Cámara Costarricense de Tecnología de Información y Comunicación*. Recuperado el 15 de Abril de 2010, de Cámara Costarricense de Tecnología de Información y Comunicación - Costa Rica: <http://www.camtic.org/ES/>

Canós, J. H., Letelier, P., & Penadé, M. C. (s.f.). *Métodologías Ágiles en el Desarrollo de Software*. Recuperado el 12 de Abril de 2010, de DSIC - Universidad Politécnica de Valencia (UPV): <http://www.willydev.net/descargas/prev/TodoAgil.pdf>

Carrillo Pérez, I., Pérez González, R., & Rodríguez Martín, A. D. (Octubre de 2008). *Metodologías de desarrollo de software*. Recuperado el 15 de Mayo de 2010, de solusoft-g11.googlecode.com/files/Metodologias%20de%20desarrollo.pdf

Carrillo Pérez, I., Pérez González, R., & Rodríguez Martín, A. D. (Octubre de 2008). *Metodologías de desarrollo de software*. Recuperado el 18 de Abril de 2010, de Solusoft-g11 Gestión de Cursos de Formación: solusoft-g11.googlecode.com/files/Metodologias%20de%20desarrollo.pdf

Carrillo Pérez, I., Pérez González, R., & Rodríguez Martín, A. D. (2008). *Metologías de Desarrollo de Software*. Recuperado el 01 de Mayo de 2010, de Solusoft-g11 Gestión de Cursos de Formación: solusoft-g11.googlecode.com/files/Metodologias%20de%20desarrollo.pdf

Casas, S. I., & Reinaga, H. H. (s.f.). *Aspectos Tempranos: Un Enfoque Basado en Tarjetas CRC*. Recuperado el 20 de Abril de 2010, de Escuela de Sistemas Universidad Nacional de Colombia: <http://pisis.unalmed.edu.co/avances/archivos/ediciones/Edicion%20Avances%202009%201/09.pdf>

Cataldi, Z. (2000). *Metodología de diseño, desarrollo y evaluación de software educativo*. Recuperado el 14 de Abril de 2010, de Centro Universitario UAEM Valle de Chalco - Universidad Autónoma del Estado de México:

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

<http://cux.uaemex.mx/cux/inv/Investigacion/OtrosTrabajos/CA/DesarrolloSoftwareEducativo.pdf>

Cavalcanti da Rocha, A. R., & Chaves Weber, K. (s.f.). *MPS.BR Lecciones Aprendidas*. Recuperado el 16 de Enero de 2010, de SOFTEX:
http://www.softex.br/mpsBr/_livros/licoes/mpsbr_es.pdf

CAVEDATOS. (2010). *Cámara Venezolana de Empresas de Tecnologías de la Información*. Recuperado el 14 de Junio de 2010, de CAVEDATOS:
<http://www.cavedatos.org.ve/index.asp>

CBTI. (2010). *Cámara Boliviana de Tecnologías de la Información*. Recuperado el 14 de Junio de 2010, de CBTI: Cámara Boliviana de Tecnologías de la Información

Centre for Software Engineering (CSE). (s.f.). *Primary Objective*. Recuperado el 18 de Febrero de 2010, de Centre for Software Engineering: <http://www.cse.dcu.ie>

Centre for Software Engineering. (s.f.). *Software Process Improvement in Regions of Europe (SPIRE)*. Recuperado el 19 de Febrero de 2010, de <http://www.cse.dcu.ie/spire/>

CEPAL. (2010). *Comisión Económica para América Latina*. Recuperado el 15 de Junio de 2010, de CEPAL: <http://www.cepal.org>

CESSI Argentina. (s.f.). *CESSI Argentina*. Recuperado el 31 de Octubre de 2009, de Cámara de Empresas de Software & Servicios Informáticos de la República Argentina: <http://www.cessi.org.ar>

CONACYT. (2010). *Consejo Nacional de Ciencia y Tecnología*. Recuperado el 14 de Junio de 2010, de CONACYT: <http://www.conacyt.gob.sv/>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Concepción Suárez, R. (2008). *MERISE - Initiation à la conception de systèmes d'information*. Recuperado el 18 de Marzo de 2010, de Comment: <http://www.commentcamarche.net/contents/merise/concintro.php3>

Concepción Suárez, R. (Octubre de 2007). *Metodología de Gestión de Proyectos en las Administraciones Públicas según ISO 10.006*. Recuperado el 18 de Marzo de 2010, de Tesis Doctorales en Xarxa (TDX): http://www.tesisenxarxa.net/TDX/TDR_UOV/TESIS/AVAILABLE/TDR-0215108-122610//UOV0024TRCS.pdf

CTIP. (2010). *Cámara de Tecnologías de la Información del Paraguay*. Recuperado el 14 de Junio de 2010, de CTIP: <http://www.ctip.org.py/v2/>

CTIP. (2010). *Estatutos sociales de la Cámara de Tecnología de la Información de Paraguay*. Recuperado el 14 de Junio de 2010, de CTIP: <http://www.ctip.org.py/v2/informacion/estatuto%20CTIP.pdf>

CUTI. (2010). *Cámara Uruguaya de Tecnologías de la Información*. Recuperado el 13 de Junio de 2010, de CUTI: <http://www.cuti.org.uy/index.php>

CYTED. (2010). *Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo*. Recuperado el 14 de Junio de 2010, de http://www.cyted.org/cyted_informacion/es/presentacion.php

De la cruz Londoño, C. A. (29 de Octubre de 2009). *Normas IEEE (Ingeniería del Software) – ISO/IEC 15404: SPICE - ISO/IEC 19770:2006*. Recuperado el 01 de Abril de 2010, de Gewins: <http://gewins.wikispaces.com/file/view/Normas+IEEE-ISO+15404-19770.pdf>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

De los Angeles, J. A. (s.f.). *Programación Orientada a Objetos*. Recuperado el 12 de Febrero de 2010, de Asociación JavaHispano:

http://www.javahispano.org/contenidos/archivo/36/intro_poo.pdf

Departamento de Informática. (s.f.). *Análisis y Diseño Orientado a Objetos - Metodologías*. Recuperado el 15 de Mayo de 2010, de Universidad de Jaén:

http://www.di.ujaen.es/asignaturas/isg/Metodologias.html#oo_methods_ooa_rd

Departamento de Sistemas Informáticos y Computación (UPV). (s.f.). *Proceso de Desarrollo de Software*. Recuperado el 15 de Mayo de 2010, de Universidad Politécnica de Valencia:

www.dsic.upv.es/asignaturas/.../lsi/.../IntroduccionProcesoSW.doc

Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. (s.f.). *ISO 9000-3*. Recuperado el 23 de Marzo de 2010, de Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia: www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/102000.doc

Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (UPV). (s.f.). *Rational Unified Process (RUP)*. Recuperado el 28 de Abril de 2010, de Portal de Desarrollo de Software, Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia (UPV):

<https://pid.dsic.upv.es/C1/Material/.../Introducción%20a%20RUP.doc>

Dolado, J. (s.f.). *REMIS*. Recuperado el 23 de Febrero de 2010, de Red sobre Experimentación y Medición en Ingeniería de Software:

<http://www.sc.ehu.es/jiwdocoj/remis/remis.htm>

El Mercado CE. (s.f.). *Definición y clasificación de las normas*. Recuperado el 05 de Abril de 2010, de Mercado CE Laboratorio de ensayos:

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

http://elmarcadoce.com/laboratorio-marcado-ce/index2.php?option=com_content&do_pdf=1&id=28

ESA. (2010). *European Space Agency*. Recuperado el 01 de Abril de 2010, de European Space Agency (ESA): <http://www.esa.int/esaCP/index.html>

Escuela de Ciencias y Sistemas Universidad de San Carlos de Guatemala. (s.f.). *UML (Unified Modeling Language)*. Recuperado el 15 de Mayo de 2010, de Scribd: <http://www.scribd.com/doc/2080534/UML>

ESI. (s.f.). *Welcome to ESI*. Recuperado el 05 de Febrero de 2010, de European Software Institute, Tecnalía: <http://www.esi.es/>

ESI@Center SINERTIC Andino, Europea Software Institute Tecnalía . (s.f.). *Modelo I.T. MARK*. Recuperado el 12 de Diciembre de 2009, de Federación Colombiana de la Industria del Software y Tecnologías Relacionadas (FEDESOFTE): http://www.fedesoft.org/downloads/Sinertic/RESUMEN_MODELO_IT_MARK.pdf

European Space Agency (ESA). (2010). *Software Engineering and Standardisation*. Recuperado el 01 de Abril de 2010, de http://www.esa.int/TEC/Software_engineering_and_standardisation/

Facultad Experimental de Ciencias y Tecnología (FACYT). (s.f.). *UML, Lenguaje de Modelado Unificado*. Recuperado el 21 de Mayo de 2010, de Facultad Experimental de Ciencias y Tecnología: <http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2006-2007/presentacion%20UML.pdf>

FEDESOFTE. (2009). *Federación Colombiana de la Industria del Software*. Recuperado el 10 de Noviembre de 2009, de Federación Colombiana de la Industria del Software: <http://www.fedesoft.org>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Fernández Enrich, M. (2003). *Crystal Methodologies*. Recuperado el 12 de Mayo de 2010, de Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia: www.dsic.upv.es/asignaturas/facultad/lisi/trabajos/282002.ppt

Fernández Lanvin, D. (2004). *Desarrollo de una Metodología para un nuevo Paradigma de Desarrollo de Software*. Recuperado el 28 de Enero de 2010, de Departamento de Informática Universidad de Oviedo: www.di.uniovi.es/~dflanvin/home/?download=pi-lanvin.pdf

Fowler, M. (2003). *La Nueva Metodología*. Recuperado el 12 de Mayo de 2010, de Programación Extrema: <http://www.programacionextrema.org/articulos/newMethodology.es.html>

Gacitúa Bustos, R. A. (2003). *Métodos de Desarrollo de Software: El desafío pendiente de la estandarización*. Recuperado el 21 de Abril de 2010, de Red de Revistas Científicas de América Latina y el Caribe, España y Portugal (Redalyc): <http://redalyc.uaemex.mx/redalyc/pdf/299/29901203.pdf>

García, M. C., Garzás, J., & Piattini, M. (s.f.). *La Mejora de Proceso en Pequeñas Empresas y la ISO/IEC 29110*. Recuperado el 05 de Febrero de 2010, de Kybele Consulting: http://www.kybeleconsulting.com/.../MCGarcia_MejoraProcesos_ISO29110.pdf

Gattaca S.A. (s.f.). *Presentación de la Metodología MSF (Microsoft Solution Framework)*. Recuperado el 12 de Mayo de 2010, de Gattaca: <http://www.e-gattaca.com/econtent/library/documents/DocNewsNo50DocumentNo6.PDF>

Gilbert Ginestá, M., & Peña González, Á. (2005). *Ingeniería del Software en Entornos SL*. Barcelona: Fundación para la Universitat Oberta de Catalunya.

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Gómez Gallego, J. P. (16 de Septiembre de 2007). *Fundamentos de la metodología RUP (Rational Unified Process) Universidad Tecnológica de Pereira*. Recuperado el 21 de Abril de 2010, de Scibd: <http://www.scribd.com/doc/297224/RUP>

Gómez García, O. (enero de 2009). *SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad*. Recuperado el 28 de Marzo de 2010, de Grupo Alarcos, Escuela Superior de Informática de Ciudad Real de la Universidad de Castilla - La Mancha: <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Oscar%20Gomez.pdf>

Gora, M. (2005). *Análisis y diseño orientados a objetos*. Recuperado el 16 de Mayo de 2010, de Universidad Autónoma de Chihuahua: <http://comunidad.uach.mx/fmarisc/analisis/LecturasSesiones/LecturaSesion04Metodologias.doc>

Granollers, T. (2004). *MPlu+a. Una Metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de Desarrollo Multidisciplinares*. Recuperado el 22 de Abril de 2010, de Tesis doctorales en red: http://www.tdr.cesca.es/TESIS_UdL/AVAILABLE/TDX-0218107-133615//Tgsa1de5.pdf

Guillén Garzaro, M. (s.f.). *Metodología para el Control de Proyectos PRINCE2*. Recuperado el 07 de Abril de 2010, de Complejo TEC de la Universidad Rafael Landívar Facultad de Ingeniería Universidad Rafael Landívar: http://www.tec.url.edu.gt/boletin/URL_09_SIS01.pdf

Gutiérrez, J. J. (s.f.). *¿Qué es un framework web?* Recuperado el 29 de Abril de 2010, de Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Hurtado, J. A., & Bastarrica, C. (2005). *Proyecto SIMEP-SW*. Recuperado el 20 de Febrero de 2010, de Departamento de Ciencias de la Computación, Universidad de Chile: http://www.dcc.uchile.cl/TR/2005/TR_DCC-2005-008.pdf

Ian Shere 2000. (s.f.). *SHERE 2000*. Recuperado el 15 de Marzo de 2010, de <http://www.shere2000.co.uk/pdf/ssadm01.pdf>

IBM. (s.f.). *IBM Rational Unified Process (RUP)*. Recuperado el 08 de Mayo de 2010, de IBM Rational Software: <http://www-01.ibm.com/software/awdtools/rup/>

IEC. (s.f.). *Bienvenidos a la IEC*. Recuperado el 15 de Febrero de 2010, de International Electrotechnical Commission: http://www.iec.ch/news_centre/onlinepubs/pdf/welcome_to_the_iec-s.pdf

IEEE. (2010). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Recuperado el 18 de Febrero de 2010, de IEEE Computer Society: <http://www.computer.org/portal/web/swebok>

IEEE. (2010). *IEEE Colombia*. Recuperado el 10 de Octubre de 2009, de The Institute of Electrical and Electronics Engineers, Inc.: www.ieee.org.co

IEEE. (2009). *IEEE Computer Society*. Recuperado el 18 de 10 de 2009, de IEEE Technical Council on Software Engineering (TCSE): <http://www.computer.org/portal/web/tcse>

IEEE. (2007). *IEEE Computer Society Colombia*. Recuperado el 16 de Enero de 2010, de IEEE Computer Society Colombia: <http://ewh.ieee.org/r9/colombia/computer/home.php>

IEEE. (2009). *IEEE Publications & Standards*. Recuperado el 18 de 10 de 2009, de IEEE Standards Association: <http://www.ieee.org/web/standards/home/index.html>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

IEEE. (2009). *IEEE Software & Systems Engineering Standards Collection*.

Recuperado el 18 de 10 de 2009, de IEEE Computer Society:

http://standards.computer.org/sesc/s2esc_geninfo/infondex.htm

IEEE. (2010). *Publications & Standards*. Recuperado el 18 de 10 de 2009, de The Institute of Electrical and Electronics Engineers, Inc:

<http://www.ieee.org/web/standards/home/index.html>

Instituto Argentino de Normalización y Certificación (IRAM). (s.f.). *Organismos de normalización*. Recuperado el 27 de Abril de 2010, de

<http://www.frsf.utn.edu.ar/iram/index.htm>

International Organization for Standardization (ISO). (2001). *ISO/IEC 9126-1 Software Engineering - Product Quality*. Ginebra.

International Organization for Standardization. (s.f.). *ISO 9001:2000*. Recuperado el 05 de Febrero de 2010, de International Organization for Standardization (ISO):

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21823

ISO 25000. (18 de Agosto de 2009). *La Norma ISO/IEC 25000:2005*. Recuperado el 28 de Marzo de 2010, de <http://iso25000.com/index.php/25000.html>

ISO. (2009). *About ISO*. Recuperado el 18 de 10 de 2009, de International Organization for Standardization : <http://www.iso.org/iso/about.htm>

ISO. (2009). *Discover ISO*. Recuperado el 18 de Octubre de 2009, de International Organization for Standardization: http://www.iso.org/iso/about/discover-iso_the-scope-of-isos-work.htm

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E
INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO. (2009). *Founding*. Recuperado el 18 de 10 de 2009, de International Organization for Standardization:

http://www.iso.org/iso/about/the_iso_story/iso_story_founding.htm

ISO IEC Information Centre. (s.f.). Recuperado el 05 de Abril de 2010, de

http://www.standardsinfo.net/info/livelihood/link/fetch/2000/148478/6301438/standards_regulations.html

ISO. (2010). *ISO 9001:2000*. Recuperado el 08 de Mayo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21823

ISO. (2010). *ISO 9241-10:1996*. Recuperado el 05 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/catalogue_detail.htm?csnumber=16882

ISO. (2010). *ISO 9241-110:2006*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38009

ISO. (2010). *ISO 9241-151:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=37031

ISO. (2010). *ISO 9241-171:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39080

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO. (2010). *ISO 9241-2:1992*. Recuperado el 27 de Febrero de 2010, de International Organization for Standardization:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16874

ISO. (2010). *ISO 9241-20:2008*. Recuperado el 06 de Marzo de 2010, de International Organization for Standardization:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40727

ISO. (2010). *ISO 9241-3:1992*. Recuperado el 27 de Febrero de 2010, de International Organization for Standardization:
http://www.iso.org/iso/catalogue_detail.htm?csnumber=16875

ISO. (2010). *ISO 9241-300:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40096

ISO. (2010). *ISO 9241-302:2008* . Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40097

ISO. (2010). *ISO 9241-303:2008* . Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40098

ISO. (2010). *ISO 9241-304:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40099

ISO. (2010). *ISO 9241-305:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40100

ISO. (2010). *ISO 9241-306:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40101

ISO. (2010). *ISO 9241-307:2008*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40102

ISO. (2010). *ISO/IEC 12207:2008*. Recuperado el 02 de Abril de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43447

ISO. (2010). *ISO/IEC 15504-1:2004*. Recuperado el 22 de Mayo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38932

ISO. (2010). *ISO/IEC 25000:2005*. Recuperado el 28 de Marzo de 2010, de International Organization for Standardization:

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35683

ISO. (2010). *ISO/IEC 90003:2004*. Recuperado el 23 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35867

ISO. (2010). *ISO/IEC TR 9126-2:2003*. Recuperado el 26 de Febrero de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22750

ISO. (2010). *ISO/IEC TR 9126-3:2003*. Recuperado el 26 de Febrero de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22891

ISO. (2010). *ISO/IEC TR 9126-4:2004*. Recuperado el 24 de Febrero de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39752

ISO. (2010). *ISO/TR 9241-100:2010*. Recuperado el 08 de Marzo de 2010, de International Organization for Standardization:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=52712

ISO. (2009). *Key markers in ISO's history*. Recuperado el 18 de octubre de 2009, de International Organization for Standardization:

http://www.iso.org/iso/about/the_iso_story.htm

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

ISO/IEC. (2010). *Standards and Regulations*. Recuperado el 01 de Abril de 2010, de ISO/IEC Information Centre:

http://www.standardsinfo.net/info/livelink/fetch/2000/148478/6301438/standards_regulations.html

ITC Infotech India Ltd. (s.f.). *NASSCOM*. Recuperado el 12 de Marzo de 2010, de <http://www.nasscom.in/download/ssad.pdf>

Laboratorio de Sistemas de Información - Universidad Politécnica de Valencia. (2010). *ISO 9000-3*. Recuperado el 02 de Mayo de 2010, de Departamento de Sistemas Informáticos y Computación UPV:

www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/102000.doc

Laboratorio de Sistemas de Información UPV. (2010). *Departamento de Sistemas Informáticos y Computación UPV*. Recuperado el 02 de Mayo de 2010, de www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/102000.doc

López Quesada, J. A. (2006). *05BM - Fundamentos de Ingeniería del Software*.

Recuperado el 18 de Abril de 2010, de Universidad de Murcia:

http://dis.um.es/~lopezquesada/documentos/FIS_0506/doc/teoria/Tema7.pdf

Madritel. (s.f.). Recuperado el 29 de Abril de 2010, de

<http://web.madritel.es/personales3/edcollado/ingsw/tema2/2-4.htm>

Martínez, A., & Martínez, R. (s.f.). *Guía a Rational Unified Process*. Recuperado el 28 de Abril de 2010, de Departamento de Sistemas Informáticos, Escuela Politécnica Superior de Albacete – Universidad de Castilla la Mancha: <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>

McGraw-Hill, Interamericana de España. (s.f.). Recuperado el 10 de Abril de 2010, de <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448146433.pdf>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Mendoza Sanchez, M. A. (07 de Junio de 2004). *Metodologías De Desarrollo De Software*. Recuperado el 12 de Abril de 2010, de Informatizate:
http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf

Mendoza, L. E., Grimán, A., & Pérez, M. (s.f.). *Especialización de MSF para el Desarrollo Basado en Componentes de un Sistema Colaborativo*. Recuperado el 21 de Mayo de 2010, de Willydev:
http://www.willydev.net/descargas/WillyDEV_Especializacion_MSF.pdf

Microsoft Corporation. (2000). *Microsoft Solutions Framework (MSF): Disciplinas y buenas prácticas para el desarrollo e implantación de proyectos*. Recuperado el 21 de Mayo de 2010, de Microsoft Colombia:
<http://www.microsoft.com/colombia/portafolio/msf.htm>

Ministerio de Administración Pública (MAP). (2010). *EUROMÉTODO v1*. Recuperado el 01 de Mayo de 2010, de Consejo Superior de Administración Electrónica de España: <http://www.csi.map.es/csi/pg5e40.htm>

Ministerio de Administración Pública (MAP). (2010). *MAGERIT – versión 2. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información*. Recuperado el 08 de Mayo de 2010, de Consejo Superior de Administración Electrónica de España: <http://www.csi.map.es/csi/pg5m20.htm>

Ministerio de Administración Pública (MAP). (2001). *MÉTRICA. VERSIÓN 3 Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información*. Recuperado el 07 de Abril de 2010, de Consejo Superior de Administración Electrónica de España: <http://www.csi.map.es/csi/metrica3/>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Ministerio de Comunicaciones de Colombia. (2009). *Planificación Programas de Apoyo a las Tecnologías de Información (PRATI)*. Recuperado el 03 de Marzo de 2009, de Ministerio de Comunicaciones: <http://www.mincomunicaciones.gov.co>

Molpeceres, A. (15 de Diciembre de 2002). *Procesos de desarrollo: RUP, XP y FDD*. Recuperado el 19 de Mayo de 2010, de Willydev: <http://www.willydev.net/descargas/articulos/general/cualxpfdrrup.PDF>

Moraga, H., & Guerrero, A. (16 de Diciembre de 2005). *Programación Ágil: SCRUM y XP*. Recuperado el 21 de Abril de 2010, de Energy Market: http://www.emarket.cl/dir/umayor/ingsw/Apoyo/GRUPO_1_PROGRAMACION_AGIL.ppt

NASSCOM. (s.f.). Recuperado el 15 de Marzo de 2010, de <http://nasscom.in/Nasscom/templates/NormalPage.aspx?id=5365>

Oktaba, H. (s.f.). *Lecciones aprendidas del desarrollo de la norma internacional ISO/IEC 29110 basada en MoProSoft*. Recuperado el 26 de Marzo de 2010, de Slidechared: <http://www.slideshare.net/RevistaSG/lecciones-aprendidas-del-desarrollo-de-la-norma-internacional-isoiec-29110-basada-en-moprosoft>

Olivares Rojas, J. C. (s.f.). *MSF*. Recuperado el 21 de Mayo de 2010, de Instituto Tecnológico de Morelia: <http://antares.itmorelia.edu.mx/~jcolivar/documents/msf.pdf>

Ortín Ibáñez, M. J., & García Molina, J. (s.f.). *Modelado basado en roles con UML*. Recuperado el 15 de Mayo de 2010, de Departamento de Informática, Lenguajes y Sistemas - Universidad de Murcia: <http://dis.um.es/~jmolina/rolesuml.pdf>

Pardo Calvache, C. J. (Enero de 2009). *Análisis comparativo del estándar ISO 9000-3 con las subcaracterísticas de calidad de la ISO 9126*. Recuperado el 22 de Febrero

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

de 2010, de Grupo Alarcos Universidad de Castilla-La Mancha:

[/doc/cmsi/trabajos/Cesar Pardo.pdf](#)

Pinzón, S., & Guevara Bolaños, J. C. (s.f.). *La gestión, los procesos y las metodologías de desarrollo de Software*. Recuperado el 08 de Abril de 2010, de Universidad Distrital Francisco José de Caldas:

<http://www.udistrital.edu.co/comunidad/dependencias/revistavinculos/VINCULOS/revista/4edicion/12006408.pdf>

Romero Guillén, P. (s.f.). *Ingeniería de Software Orientada a Objetos OOSE (IVAR JACOBSON)*. Recuperado el 14 de Mayo de 2010, de Instituto Tecnológico de la Laguna:

<http://www.itlalaguna.edu.mx/Academico/Carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/Jacobson.pdf>

Romero Guillén, P. (s.f.). *Métodos y Modelos, Análisis Orientado a Objetos*.

Recuperado el 02 de Mayo de 2010, de Instituto Tecnológico de la Laguna:

<http://www.itlalaguna.edu.mx/Academico/Carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/cap2.pdf>

Santos, G., Chaves Weber, K., & Cavalcanti da Rocha, A. R. (2009). *SOFTEX*.

Recuperado el 10 de Noviembre de 2009, de

http://www.softex.br/portal/softexweb/uploadDocuments/clei2009_mpsbr%20030809%281%29.pdf

Saravia Aramayo, R. (Noviembre de 2007). *Estándar ISO/IEC-14598 - Evaluación del Producto Software*. Recuperado el 05 de Abril de 2010, de Scribd:

<http://www.scribd.com/doc/14897492/Estandar-ISOIEC14598-Evaluacion-del-Producto-Software>

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SIGSOFT. (s.f.). *The ACM Special Interest Group on Software Engineering* .

Recuperado el 28 de Abril de 2010, de <http://www.sigsoft.org/>

Sociedad Chilena de Software y Servicios A.G. (GECHS). (s.f.). *Estatutos Sociedad Chilena de Software y Servicios A.G.* Recuperado el 14 de Noviembre de 2009, de GECHS Software y Servicios Chile A.G:

<http://bligoo.com/media/users/0/32814/files/Estatutos-Asamblea%20General.pdf>

Sociedad Chilena de Software y Servicios A.G. (GECHS). (s.f.). *Historia*. Recuperado el 14 de Noviembre de 2009, de GECHS Software y Servicios Chile A.G:

<http://www.gechs.cl/content/view/32439/Historia.html>

Sociedad SOFTEX. (s.f.). *MPS.BR - Mejora de Proceso del Software Brasileño. Guía de Adquisición*. Recuperado el 08 de Febrero de 2010, de SOFTEX:

http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Gu%C3%ADa_de_Adquisici%C3%B3n_2009.pdf

Sociedad SOFTEX. (s.f.). *MPS.BR - Mejora de Proceso del Software Brasileño. Guía General*. Recuperado el 08 de Febrero de 2010, de SOFTEX:

http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Gu%C3%ADa_General_2009.pdf

SOFEX. (s.f.). Recuperado el 19 de Diciembre de 2009, de

<http://www.export.com.gt/Portal/Home.aspx?secid=1464>

SOFTEX. (s.f.). Recuperado el 10 de Noviembre de 2009, de

http://www.softex.br/_asoftex/sociedadeSoftex.asp

SOFTEX. (s.f.). Recuperado el 10 de Noviembre de 2009, de

http://www.softex.br/_asoftex/diretrizesEstrategicas.asp

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

SOFTEX. (s.f.). Recuperado el 08 de Febrero de 2010, de http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Gu%C3%ADa_de_Evaluaci%C3%B3n_2009.pdf

System Concepts. (2009). *System Concepts*. Recuperado el 11 de Mayo de 2010, de Usability, Ergonomics, Health and Safety Consultants: <http://www.system-concepts.com/articles/usability-and-ergonomics-standards/three-new-accessibility-and-web-design-standards-iso-9241-parts-20-151-and-171.html>

The IEEE Computer Society. (s.f.). Recuperado el 16 de Enero de 2010, de www.computer.org

The International Software Benchmarking Standards Group Limited . (s.f.). *ISBSG*. Recuperado el 18 de Febrero de 2010, de The International Software Benchmarking Standards Group Limited : <http://www.isbsg.org>

The Software Engineering Technical Committee of the Computer Society of the IEEE. (1988). *IEEE Standard for Software Project Management Plans*. Recuperado el 02 de Abril de 2010, de Kybele - Grupo de investigación de la Universidad Rey Juan Carlos: <http://www.kybele.etsii.urjc.es/docencia/IS3/2009-2010/Material/%5BIS3-0910%5DIEEE-1058.1.pdf>

Travis, D. (s.f.). *Bluffers' Guide to ISO 9241*. Recuperado el 25 de Febrero de 2010, de USERFOCUS: <http://www.userfocus.co.uk/pdf/ISO9241.pdf>

Universidad de Huelva. (s.f.). *El lenguaje unificado de modelado, UML*. Recuperado el 21 de Mayo de 2010, de Universidad de Huelva: http://www.uhu.es/juanmanuel.cordoba/old/d_is2/teoria/tema1/t1a.pdf

ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA

Ingeniería de Sistemas y Telecomunicaciones
Universidad Católica Popular del Risaralda

Universidad de Huelva. (s.f.). *El lenguaje unificado de modelado, UML*. Recuperado el 21 de Mayo de 2010, de Universidad de Huelva:

http://www.uhu.es/juanmanuel.cordoba/old/d_is2/teoria/tema1/t1a.pdf

Universidad del Cauca. (2006). *COMPETISOFT*. Recuperado el 2002 de 14 de 2010, de Universidad del Cauca:

http://afrodita.unicauca.edu.co/~ecaldon/docs/spi/COMPETISOFT_v02_27-11_2315.pdf

University of Glamorgan. (2010). *Introduction to Methodologies and SSADM*.

Recuperado el 28 de Marzo de 2010, de University of Glamorgan:

<http://www.comp.glam.ac.uk/pages/staff/tdhutchings/chapter4.html>

Usability Net. (s.f.). *ISO 9241: Ergonomic Requirements for Office work for Display Terminals*. Recuperado el 27 de Febrero de 2010, de Usability Net:

http://www.usabilitynet.org/tools/r_international.htm#9241-1x