

ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS EMPLEADAS PARA EL DISEÑO DE PROCESOS



GRUPO DE INVESTIGACIÓN: GIS
PROYECTO DE INVESTIGACIÓN: METAMODELO PARA EL CÁLCULO DEL ESFUERZO EN
EL DESARROLLO DEL SOFTWARE

INVESTIGADOR: JORGE ENRIQUE OTÁLORA LUNA ¹

INVESTIGADOR: JUAN JOSÉ CAMARGO VEGA ²

INVESTIGADORA: AURA BEATRIZ ALVARADO GAONA ³

RESUMEN

El presente artículo, muestra un estudio comparativo de las herramientas empleadas para el diseño de procesos, teniendo en cuenta sus características y ventajas principales, que ayudarán a establecer cuál de éstas tiene un mejor desempeño y brinda mayores prestaciones, al momento de utilizarlas.

Este trabajo hace parte de investigación "Metamodelo para el cálculo del esfuerzo en el desarrollo de software", que aportará el medio de selección apropiado de acuerdo a las características relevantes que determinan el estudio.

PALABRAS CLAVE

Diseño de procesos, metamodelo, Herramientas de Software.

ABSTRACT

This article shows a comparative study of the tools used to process modeling, taking into account its main features and benefits that will help establish which of these has a better performance and provides greater benefits when using them.

This work is part of research project "metamodel for calculation effort in the software development" that will be the middle appropriate according to the relevant characteristics that determines the study.

KEY WORDS

Process design, metamodel, software tools.

Fecha de recepción del artículo: Agosto 21 de 2009

Fecha de aceptación del artículo: Septiembre 24 de 2009

- 1 Profesor Auxiliar, Facultad de Ingeniería, Universidad Pedagógica y Tecnológica de Colombia - Tunja - Colombia. Especialista en Ingeniería de Software - Universidad Antonio Nariño - Bogotá - Colombia. Estudiante de Doctorado en Informática - Universidad Pontificia de Salamanca - Madrid - España.
- 2 Profesor Asistente, Facultad de Ingeniería, Universidad Pedagógica y Tecnológica de Colombia - Tunja - Colombia. Magíster en Tecnología de la Información Aplicadas a la Educación - Universidad Pedagógica Nacional - Bogotá - Colombia. Estudiante de Doctorado en Informática - Universidad Pontificia de Salamanca - Madrid - España.
- 3 Docente Investigador, Programa Ingeniería de Sistemas, Facultad de Ingeniería, Integrante Grupo de Investigación DAVINCIS, Universidad Libre Colombia - Bogotá - Colombia. Doctorando en Ingeniería Informática - Universidad Pontificia de Salamanca - Madrid - España.

INTRODUCCIÓN

Este artículo, presenta un estudio comparativo de algunas de las herramientas empleadas para el modelado de procesos, que se logró, luego de una revisión teórica, el diseño de métricas y la aplicación de las mismas de forma práctica, teniendo en cuenta sus características principales, funcionalidad y estandarización.

La comparación se realizó debido a la importancia que representa la adecuada selección de las herramientas, para obtener el éxito en cualquier tipo de proyecto de desarrollo de software, y al aporte que representa dentro de cada uno de los proyectos de investigación en los que se encuentran trabajando los autores del presente artículo, así:

- “Metamodelo para el cálculo del esfuerzo en el desarrollo de software” de Jorge Enrique Otálora Luna, allí es necesario conocer las herramientas de modelado especialmente desde el punto de vista de la interoperabilidad que ofrecen y de los estándares de modelado en los que se basan.
- "Metamodelo para Integración de Modelos de Procesos de Negocio" de Aura Beatríz Alvarado Gaona, en dónde la comunicación de diferentes tipos de herramienta, es fundamental para lograr el objetivo propuesto, además de conocer las características primordiales a la hora de representar modelos de negocio.
- “Utilizar la arquitectura orientadas al servicio y gestión de procesos de negocios en la gestión de la calidad del software”, de Juan José Camargo Vega, lo cual necesita para lograr su realización conocer el nivel de calidad de las herramientas de modelado de procesos.

Por los aportes que representa esta temática común, los autores deciden realizar un experimento conjunto, que da como resultado un cuadro comparativo de las

principales características de las herramientas de modelado, así como una metodología para su evaluación con sus respectivas métricas.

A través del desarrollo del presente artículo se podrá encontrar, la fundamentación teórica que soporta el estudio realizado, la metodología que se llevó a cabo para lograrlo, los resultados obtenidos luego de la aplicación de la misma, así como las conclusiones y recomendaciones derivadas de la comparación.

1. MARCO TEÓRICO

A continuación se presenta la fundamentación teórica utilizada por los autores para la realización del presente artículo.

1.1 El papel de los metamodelos y entornos para el modelado de procesos

Se denomina metamodelo de procesos, al modelo de un lenguaje de modelado, que describe un conjunto de conceptos genéricos y sus interrelaciones, los cuales sirven de base para la definición de modelos de procesos. (Meta Object Facility) (Pavlich, 2005) (Favre, 2005) (Iván García-Magariño); dicho de otra forma, el metamodelo debe abstraer de los modelos que se pueden instanciar en la realidad tanto los elementos estáticos como los dinámicos, de tal manera que se permita la creación de modelos nuevos que tengan en cuenta las características genéricas y las adecúen al ámbito según los requisitos o necesidades específicas.

Una de las grandes ventajas que se consiguen con la utilización de los metamodelos es la resolución de problemas en el ambiente del paradigma en el que se encuentra, gracias a la creación o derivación de modelos específicos. Por otra parte, hay que tener en cuenta que los metamodelos impiden que se pueda observar más allá de sus fronteras, por lo que es necesario comprender la dinámica del metamodelo, haciendo una redefinición constante para evitar su estancamiento.

Al momento de realizar la definición de metamodelos, se requiere la utilización de alguno de los lenguajes existentes, tales como: Meta Object Facility (MOF) especificado por Object Management Group (OMG) (OMG), GOPRR (Graph Object Property Relationship Role) (Kelly, 1997) y Ecore, lenguaje empleado por el Eclipse Modeling Framework (EMF) (Budinsky, 2006).

1.1.1 Utilización de los metamodelos

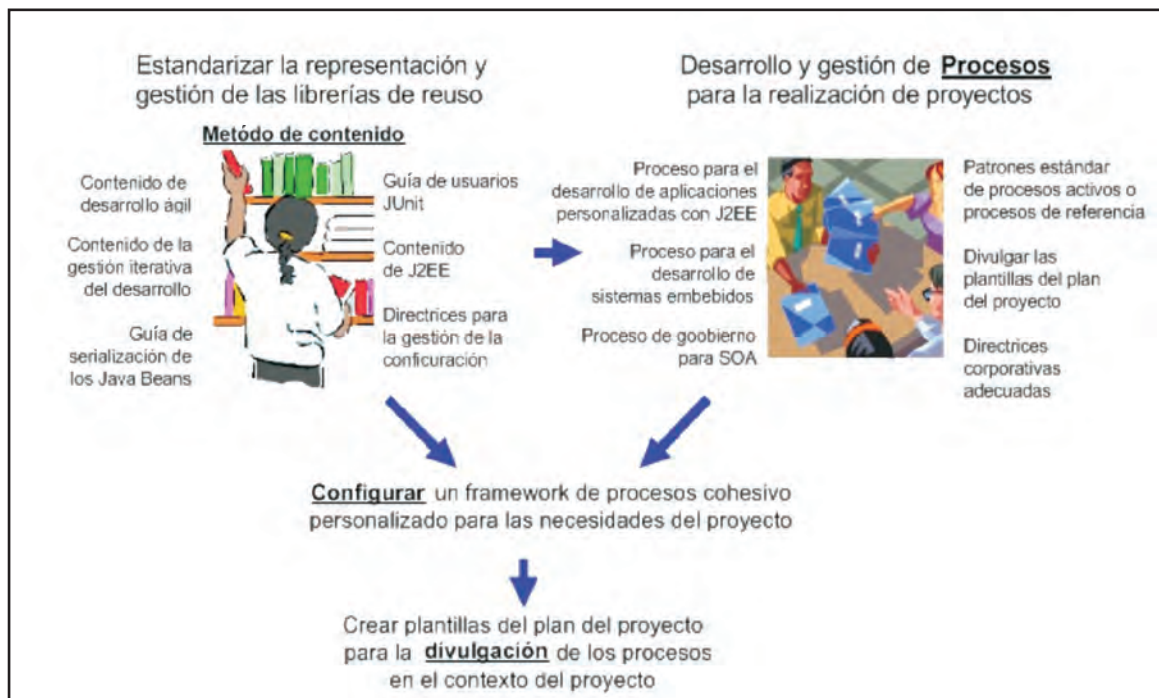
Los metamodelos son utilizados por UML para realizar la definición de la sintaxis abstracta de UML y con base en ésta definir sintaxis concretas, algunos ejemplos se enuncian a continuación (Universidad Rey Juan Carlos).

SPEM (Software Process Engineering Metamodel), es un metamodelo empleado para definir modelos de procesos. (Félix García). Éste brinda un marco de trabajo adecuado, para realizar el modelado, documentación, presentación, gestión e intercambio de los procesos de desarrollo Software y sus componentes. De igual manera SPEM facilita la integración del proceso de software al permitir asignar a cada uno de las partes del proceso de desarrollo, una estructura común que incluye roles, tareas y guías (Larrucea, Juncal, Garcia, & Barcelona, 2007).

Metamodelo UPM (Unified Process Model). Es un modelo conceptual que tiene como fin principal definir el grupo de elementos de modelado que se emplean para diseñar el Modelo de Procesos RUP. (Rational Unified Process) (Montlva).

Figura 1

Marco de trabajo conceptual de SPEM.
(Larrucea, y otros, 2007)

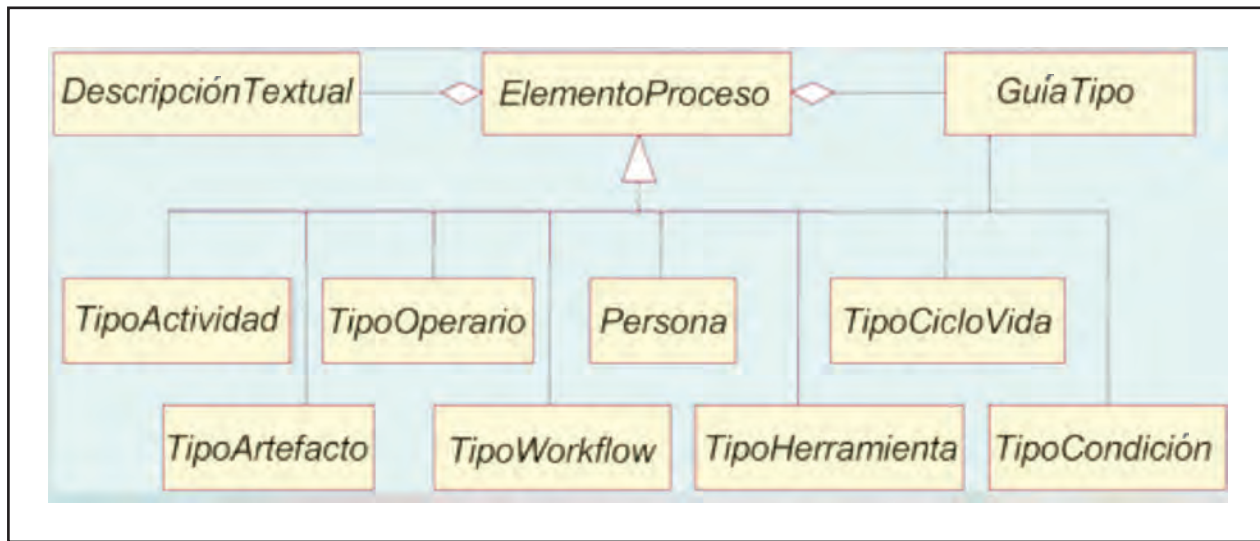


Fuente: Larrucea, Xabier. Especificación del metamodelo a utilizar [en línea]. [Consultado 06 de Junio de 2009] Disponible en Internet:

www.ines.org.es/.../d2-1-especificacion-del-metamodelo-spem-2-0-v10.doc

Figura 2

Estructura del metamodelo UPM.



Fuente: MONTILVA, Jonás A. Modelado de procesos de Software [en línea]. Merida [consultado 15 de Abril de 2009]. Disponible en Internet: <http://alarcos.inf-cr.uclm.es/doc/psgc/doc/lec/parte2b/SPE-IDEA.pdf>.

1.1.2 Modelo de Procesos

Los modelos de procesos básicamente son una abstracción de la realidad, que buscan captar los aspectos de mayor importancia de un proceso de software. Éstos describen las actividades, recursos, actores, reglas y productos necesarios para lograr los objetivos.

La construcción de los modelos de procesos se realiza mediante la instanciación de los conceptos de un metamodelo de procesos. Dicha instanciación se determina principalmente por las características propias del modelo que se desea elaborar, entre las que se encuentran los objetivos, dominio de aplicación, entorno, entre otras.

Al momento de realizar el diseño del modelo de procesos es importante conservar las relaciones entre los conceptos definidos anteriormente en el metamodelo. (Montlva).

1.2 Tipos de Software

En la industria informática existen diferentes categorías que enmarcan la naturaleza del software, depende del tipo de licencia que emplee para el mismo. Cada una de ellas están sujetas a diferentes características o restricciones de distribución y uso tal como se enuncia a continuación:

1.2.1 Software libre

Este tipo de software, permite la libre adquisición, modificación y distribución de los programas. Se distribuye bajo la licencia GPL (General Public License), la cual se mantiene, aún cuando el usuario haya realizado modificaciones al mismo. Esta distribución incluye tanto el programa como el respectivo código fuente.

1.2.2 Freeware

Este tipo de software, se adquiere de manera gratuita pero está limitado tanto el tiempo de uso como la

funcionalidad de la herramienta. No permite la modificación del código y contempla algunas restricciones para permitir su redistribución.

1.2.3 Software Propietario

Hace referencia al software sobre el que una persona o compañía conserva los derechos de autor, restringiendo el uso, modificación y adquisición de la herramienta, así como el acceso al código fuente.

2. METODOLOGÍA

Para el desarrollo del estudio comparativo, fue necesario realizar las actividades que se enuncian a continuación, las cuales permitieron identificar la herramienta que mejor se ajusta a las necesidades y requisitos de los proyectos a realizar por los autores.

Recopilación - clasificación de herramientas de modelado de procesos. En esta primera actividad, se llevó a cabo una búsqueda de las herramientas más utilizadas para realizar el modelado de procesos. Una vez se obtuvo la recopilación, se clasificaron de acuerdo al tipo de licencia o software al que pertenecen.

Descripción de las herramientas de modelado de procesos. Se realizó una nueva búsqueda en profundidad, que permitió obtener un alto nivel de comprensión y claridad en cuanto a la usabilidad, manejo y ventajas de dichas herramientas. Con la información obtenida se redactó la descripción de cada una basado en sus características generales.

Definición de las características a evaluar en las herramientas. Luego de examinar las herramientas, se escogieron las características relevantes que debe poseer cada una para ser utilizada en los proyectos. Esta actividad es de gran importancia debido a que la

adecuada selección de las mismas permitirá alcanzar el éxito del trabajo propuesto.

Asignación de pesos a las características. De acuerdo a la relevancia e importancia de cada una de las características elegidas en el paso anterior, se procedió a asignarles peso, el cual permite realizar una evaluación cuantitativa de las herramientas, y según los resultados obtenidos elegir la que brinde mejores prestaciones y utilidad al proyecto.

Evaluación de las herramientas. Para realizar la evaluación de las herramientas se elaboró un cuadro comparativo, que permitió indicar el peso correspondiente según la característica o características presentes en cada caso. Una vez asignados los valores, se totalizaron los resultados.

Selección de la herramienta. La herramienta que se escogerá será la que alcance una mayor puntuación según las características elegidas y los pesos asignados en cada caso, y según lo que sea más significativo en cada uno de los proyectos.

3. RESULTADOS

Para realizar el estudio comparativo, se llevaron a cabo las actividades planteadas en la metodología las cuales permitieron la obtención de los resultados que se muestran a continuación:

3.1 Recopilación - clasificación de herramientas modelado de procesos

De acuerdo con la investigación realizada, se encontraron algunas de las herramientas de modelado de procesos disponibles en el mercado, clasificadas acorde al tipo de software en el que se encuentran enmarcadas (wikipedia, 2009):

Tabla 1

Clasificación de las herramientas de modelado según el tipo de software que manejan.

TIPO DE SOFTWARE	HERRAMIENTAS
Software Libre	ArgoUML, BOUML, Fujaba, Dia, gModeler, MonoUML, Papyrus, StarUML, TCM, Toolkit for Conceptual Modeling, Umbrello, UMLet, Netbeans módulo UML, Open ModelSphere
Freeware	JUDE Community, Omondo plugin para Eclipse, Oracle JDeveloper, Visual Paradigm for UML
Software Propietario	Enterprise Architect de Sparx Systems, Borland Together, Corel iGrafx, Microsoft Visio, PowerDesigner de Sybase, Rational Rose de IBM, Poseidon for UML de GentleWare, MagicDraw UML.0

3.2 Descripción de las herramientas modelado de procesos

Acorde al listado recopilado, se realizó una breve descripción de cada una de las herramientas, con el fin de tener un panorama amplio de algunas de sus

características y utilidades. La información obtenida se organizó en forma tabular, como se muestra a continuación:

Tabla 2

Herramientas para el modelado de procesos.

SOFTWARE LIBRE	
ArgoUML	Herramienta desarrollada en Java bajo licencia BSD Open Source, empleada para el diseño de diagramas UML. Puede utilizarse en cualquier sistema operativo que soporte java (Odutola, y otros, 2008).
BOUML	Herramienta multiplataforma, utilizada para el diseño y creación de diagramas UML. Permite generar código en IDL, C++ y php de forma automática, a través de los modelos realizados.(BOUML, 2009).
Fujaba	Herramienta CASE, que permite la realización de diagramas UML, para el modelado de diferentes aplicaciones, y a la vez brinda la posibilidad de generar código java de manera automática. La ingeniería a la inversa es otra de las funcionalidades de esta herramienta (Fujaba, 2007).
Dia	Herramienta de modelado que además de permitir la realización de diagramas UML, se puede emplear para crear diagramas para circuitos, cronogramas, redes entre otros (Capítulo de Estudiantes de ACM, 2003).

SOFTWARE LIBRE	
gModeler	Herramienta basada en flash, realizada por Grant Skinner, que permite la creación de modelos UML. De igual manera, con esta herramienta es posible generar código Action Scrip 2.0(Skinner, 2003).
MonoUML	Herramienta CASE, empleada para el diseño y creación de diagramas UML. Permite generar código C# a partir de los modelos UML. Fue realizada para la plataforma Mono(Baeza, 2005).
Papyrus	Herramienta basada en Eclipse, que permite la realización de diagramas empleando el modelado UML 2(Papyrus, 2008).
StarUML	Herramienta desarrollada en Delphi, empleada para el modelado UML. Permite realizar autogeneración de código (Java, C++, C#...) entre otros(StarUML).
TCM, Toolkit for Conceptual Modeling	Herramienta de modelado que permite la realización de diversos tipos de diagramas, incluyendo aquellos que corresponden a UML. TMC fue creado bajo licencia GNU y corre sobre plataformas Linux, Unix, Windows(TCM, 2003).
Umbrello	Herramienta libre desarrollada por Paul Hensgen, que permitie crear y editar diagramas UML en entornos KDE. Esta herramienta genera código automáticamente en los lenguajes C++, Java, Python, IDL, Pascal/Delphi, Ada (Umbrello, 2008).
UMLet	Herramienta que emplea una interface ligera para permitir la creación de diagramas UML de forma fácil y rápida. Fue escrita en java y se puede utilizar como una aplicación independiente o integrada en Eclipse, como un plugin (Navarro, 2005).
Netbeans módulo	Este módulo permite realizar el modelado de diferentes aplicaciones empleando UML. Incluye la funcionalidad de generación automática.
UML	Código para Java, C++, php entre otros(NetBeans).
Open ModelSphere	Herramienta libre, creada en lenguaje Java, que permite la realización de diagramas UML. Soporta Ingeniería a la inversa de una base de datos SQL via ODBC y JDBC(Open Model Sphere, 2009).

FREWARE	
JUDE Community	Herramienta de modelado que incluye características básicas para la creación y edición de diagramas UML(change-vision, 2006).
Omondo plugin para Eclipse	Herramienta de modelado que permite la creación y edición de diagramas UML. Se integra en el entorno eclipse y contiene la funcionalidad de Ingeniería a la inversa. Se puede emplear en varios sistemas operativos tales como Linux, Windows, Solaris 8 y MacOS (Omondo, 2009).

FREWARE	
JUDE Community	Herramienta de modelado que incluye características básicas para la creación y edición de diagramas UML(change-vision, 2006).
Omondo plugin para Eclipse	Herramienta de modelado que permite la creación y edición de diagramas UML. Se integra en el entorno eclipse y contiene la funcionalidad de Ingeniería a la inversa. Se puede emplear en varios sistemas operativos tales como: Linux, Windows, Solaris 8 y MacOS (Omondo, 2009).
Oracle Jdeveloper	Es un IDE para SOA y Java que permite soporta la realización de diagramas UML (Oracle).
Visual Paradigm for UML	Herramienta de modelado UML y herramienta CASE que permite realizar diagramas UML, generar código y documentación a partir de dichos diagramas y realizar ingeniería a la inversa. Existe una versión gratuita llamada Community Edition (Visual Paradigm).

SOFTWARE PROPIETARIO	
Enterprise Architect de Sparx Systems	Esta herramienta de modelado, se basa en el estándar de UML 2.1 y ayuda en la realización de diferentes actividades que se llevan a cabo durante el desarrollo de software (Sparx Systems Pty Ltd, 2008).
Borland Together	Es una completa herramienta para modelado, que permite realizar el diseño de aplicaciones, desarrollo y despliegue, facilitando al equipo de trabajo la planeación y construcción de sistemas robustos (Borland®, 2009).
Corel iGrafx	Es una completa suite que permite, modelar, diseñar y mejorar los procesos de negocio en una organización. Se pueden encontrar en diferentes ediciones de acuerdo a las necesidades de la organización (Corel Corporation, 2009).
Microsoft Visio	Es una herramienta que permite la elaboración de diversos diagramas de oficina, bases de datos, diagramas de flujo de programas, UML, entre otros. Se encuentra disponible en dos versiones: Office Visio Professional y Office Visio Standard (Microsoft Corporation, 2009).
Power Designer de Sybase	Es una herramienta para modelado, considerada como una de las mejores del mercado. Entre sus características principales se pueden citar, la fácil visualización, análisis, diseño y manipulación de metadatos, que garantizan una arquitectura empresarial efectiva (2008).

SOFTWARE PROPIETARIO	
Rational Rose de IBM	Es una herramienta para modelado, que soporta UML 2.0, permite la creación de diferentes diagramas, así como la generación automática de código a diferentes lenguajes tales como: Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic (IBM, 2009).
Poseidon for UML de GentleWare	Es una herramienta de modelado, que permite realizar diagramas de forma rápida y sencilla, crear documentación y generar código de manera automática. Contiene una interfaz intuitiva que facilita su comprensión y manejo (Gago, 2006).
MagicDraw UML	Esta herramienta para modelado UML, permite realizar diferentes diagramas y soporta varios lenguajes de programación tales como: C, C++, .NET, Schema, CORBA. De igual manera, facilita la creación de Bases de Datos y sistemas orientados a objetos (Magic Draw, 2009).

3.3 Definición de las características a evaluar en las herramientas

Para la elaboración del cuadro comparativo de las herramientas de modelado de software se tuvieron en cuenta las características que se mencionan a continuación.

3.3.1 Categoría

Hace referencia al tipo de software de acuerdo a su licenciamiento. La clasificación se encasilla en herramientas de carácter libre, propietario y freeware, teniendo en cuenta las ventajas, desventajas y políticas de uso y distribución de cada una de ellas, en las etapas de desarrollo y mercadeo.

3.3.2 Plataforma

Teniendo en cuenta que las empresas PyMEs manejan diferentes plataformas para el desarrollo y puesta en marcha de sus productos, se considera la posibilidad de que la propuesta desarrollada pueda ser empleada en múltiples plataformas.

3.3.3 Generación de código

Durante el desarrollo de la propuesta, se puede llegar a necesitar la generación de código fuente para ser exportado y usado en otras herramientas, lo que podría llegar a facilitar el establecimiento de la interoperabilidad entre las herramientas.

3.3.4 Soporte de ingeniería inversa

Se considera la posibilidad de necesitar la importación de código fuente y a partir de este generar diagramas.

3.3.5 Soporte a UML

Teniendo en cuenta que UML, es un lenguaje estándar para el modelado de diagramas, respaldado por el OMG, se requiere que la herramienta seleccionada tenga soporte a este tipo de diagramas. La solución desarrollada permitirá entonces que las empresas compitan en un mercado global y que cualquier empresa pueda hacer uso de la propuesta desarrollada.

3.3.6 Estabilidad de la herramienta

Se requiere hacer uso de una herramienta cuyo funcionamiento sea estable, a fin de evitar inconvenientes durante el desarrollo y uso de la propuesta desarrollada.

3.3.7 Soporte al estándar xmi/xpdl

Al trabajar con herramientas que soporten estándares, se garantiza su usabilidad por parte de diferentes organizaciones y que éstas cuenten con un fuerte que facilite la competitividad en el mercado.

3.3.8 Depurador de diseño

Aunque no constituye una característica primordial, aporta valor agregado, porque se podría verificar el diseño y funcionalidad de un proceso antes de su planeación y ejecución.

3.3.9 Diagramas UML soportados

Se hace una revisión de las herramientas en cuanto al soporte que brindan en la generación de diagramas de casos de uso, clases, estados, actividad, colaboración, secuencia, componentes y despliegue, soportados por UML.

3.3.10 Enlace a la página principal de la herramienta

Se considera este ítem adicional, para acceder a la documentación oficial y descarga o compra del software.

3.4 Asignación de pesos a las características

Acorde a las características seleccionadas, se asigna un valor de importancia a cada una de ellas y se tienen en

cuenta criterios de evaluación, a fin de facilitar la elección de la herramienta a utilizar. Las tablas con las consideraciones para la evaluación de las herramientas se muestran a continuación.

Para realizar la escala de evaluación se consideraron tres rangos a los que se les asignó un peso, de acuerdo a su importancia de mayor a menor equivalencia como se muestra en la tabla:

- **Imprescindible:** son las características que obligatoriamente debe incluir la herramienta para poder ser usada en el desarrollo del proyecto.
- **Necesario:** son las características que brindan funcionalidad al sistema y de las que no se puede prescindir.
- **Complementario:** son características adicionales de las que se puede prescindir.

Tabla 3

Escala de Evaluación de las herramientas.

ESCALA DE EVALUACIÓN	PESO
Imprescindible	3
Necesario	2
Complementario	1

Una vez se estableció la escala de evaluación, se procedió asignar un peso a cada una de las características para tener en cuenta de acuerdo a su importancia en el desarrollo de este proyecto.

Tabla 4

Asignación de peso a las características a evaluar.

CARACTERÍSTICA EVALUADA	PESO ASIGNADO
Categoría	1
Plataforma	2
Generación de código	1
Soporte de ingeniería inversa	1
Soporte a UML	3
Estabilidad de la herramienta	2
Soporte al estándar XMI	3
Depurador de diseño	1

A cada característica elegida, se le asignaron criterios de evaluación acorde a los valores que podía tomar cada una de ellas, y se estableció una ponderación según la asignación de pesos realizada previamente.

Tabla 5

Ponderación de los criterios de evaluación.

CARACTERÍSTICA EVALUADA	CRITERIOS DE EVALUACIÓN				PONDERACIÓN
Categoría	L	F	P	D	8%
	63	18	14	5	
Plataforma	Multiplataforma	Multiplataforma	D		15%
	78	17	5		
Generación de código	Java	Otros	D		5%
	68	27	5		
Soporte de ingeniería inversa	Si	No	D		5%
	78	17	5		
Soporte a UML	Si	No	D		20%
	88	7	5		
Estabilidad de la herramienta	Si	No	D		15%
	73	22	5		
Soporte al estándar XMI	Si	No	D		30%
	93	2	5		
Depurador de diseño	Si	No	D		2%
	48	47	5		

3.5 Evaluación de las herramientas

De acuerdo con los pesos asignados a las características, se realiza un cuadro de evaluación de las herramientas, en donde el peso mayor corresponde a la herramienta seleccionada.

Se puede observar la valoración que se le asignó a cada una de las características: categoría, plataforma, generación de código, ingeniería inversa, soporte a UML, estabilidad de la herramienta, soporte a los estándares XMI y depurador de diseño.

Tabla 6
Cuadro comparativo herramientas
modelado de procesos.

CARACTERÍSTICAS HERRAMIENTA	CATEGORÍA	PLATAFORMA	GENERACIÓN DE CÓDIGO	INGENIERÍA INVERSA	SOPORTE UML	EST	S. XMI/XPDL	DEP. DE DISEÑO	Total
ArgoUML	63	78	68	78	88	22	93	48	70,07
	5,04	11,7	3,4	0,17	17,6	3,3	27,9	0,96	
Borland Together	14	78	27	78	88	73	93	5	70,7875
	1,12	11,7	1,35	0,0675	17,6	10,95	27,9	0,1	
BOUML	63	78	27	78	88	73	93	47	75,5475
	5,04	11,7	1,35	0,0675	17,6	10,95	27,9	0,94	
Corel iGrafx	14	17	27	17	88	73	93	48	62,4975
	1,12	2,55	1,35	0,0675	17,6	10,95	27,9	0,96	
Dia	63	78	5	17	88	5	5	5	36,9525
	5,04	11,7	0,25	0,0125	17,6	0,75	1,5	0,1	
Enterprise Architect	14	78	68	78	88	73	93	48	73,8
	1,12	11,7	3,4	0,17	17,6	10,95	27,9	0,96	
Fujaba	63	78	68	78	88	73	2	47	50,4
	5,04	11,7	3,4	0,17	17,6	10,95	0,6	0,94	
gModeler	63	78	27	17	88	5	5	5	38,1075
	5,04	11,7	1,35	0,0675	17,6	0,75	1,5	0,1	
JUDE Community	18	78	68	78	88	73	93	5	73,26
	1,44	11,7	3,4	0,17	17,6	10,95	27,9	0,1	
MagicDraw UML	14	5	5	17	88	5	5	5	22,0825
	1,12	0,75	0,25	0,0125	17,6	0,75	1,5	0,1	
Microsoft Visio	14	17	27	17	88	73	93	5	61,6375
	1,12	2,55	1,35	0,0675	17,6	10,95	27,9	0,1	
MonoUML	63	17	5	78	88	5	93	5	54,2025
	5,04	2,55	0,25	0,0125	17,6	0,75	27,9	0,1	
Netbeans módulo UML	63	78	68	78	88	73	5	5	50,46
	5,04	11,7	3,4	0,17	17,6	10,95	1,5	0,1	

CARACTERÍSTICAS HERRAMIENTA	CATEGORÍA	PLATAFORMA	GENERACIÓN DE CÓDIGO	INGENIERÍA INVERSA	SOPORTE UML	EST	S. XMI/XPDL	DEP. DE DISEÑO	Total
Omondo plugin para Eclipse	18	5	5	17	88	5	5	5	22,4025
	1,44	0,75	0,25	0,0125	17,6	0,75	1,5	0,1	
Open Model Sphere	63	78	68	78	88	5	5	5	40,26
	5,04	11,7	3,4	0,17	17,6	0,75	1,5	0,1	
Oracle JDeveloper	18	5	5	17	88	5	5	5	22,4025
	1,44	0,75	0,25	0,0125	17,6	0,75	1,5	0,1	
Papyrus	63	78	68	78	88	73	5	5	50,46
	5,04	11,7	3,4	0,17	17,6	10,95	1,5	0,1	
Poseidon community for UML	14	78	68	17	88	73	93	5	72,94
	1,12	11,7	3,4	0,17	17,6	10,95	27,9	0,1	
PowerDesigner	14	5	5	78	88	5	5	5	22,0825
	1,12	0,75	0,25	0,0125	17,6	0,75	1,5	0,1	
Rational Rose de IBM	14	78	68	78	88	73	5	5	46,54
	1,12	11,7	3,4	0,17	17,6	10,95	1,5	0,1	
StarUML	63	17	68	17	88	73	93	5	67,71
	5,04	2,55	3,4	0,17	17,6	10,95	27,9	0,1	
TCM Toolkit for Conceptual Modeling	63,00	5	5	17	88	5	5	5	26,0025
	5,04	0,75	0,25	0,0125	17,6	0,75	1,5	0,1	
Umbrello	63	17	5	78	88	5	5	5	27,8025
	5,04	2,55	0,25	0,0125	17,6	0,75	1,5	0,1	
UMLet	63	78	68	20	88	5	5	5	40,26
	5,04	11,7	3,4	0,17	17,6	0,75	1,5	0,1	
Visual Paradigm for UML	18	78	5	80	88	5	5	5	33,3525
	1,44	11,7	0,25	0,0125	17,6	0,75	1,5	0,1	

3.6 Selección de la herramienta

De acuerdo al cuadro de evaluación generado anteriormente, se eligieron las tres herramientas con mayor ponderación, como se muestra a continuación:

BOUML	75,5475
Enterprise Architect	73,8
JUDE Community	73,26

Una vez seleccionadas las herramientas, se procedió a analizar cada una con el fin de determinar cuál de éstas brinda mejores prestaciones para el desarrollo de los proyectos. El análisis efectuado se muestra a continuación:

BOUML: es una herramienta libre, que soporta los principales lenguajes orientados a objetos, permite integrar herramientas externas que hayan sido

programas en Java o C++, existe amplia documentación en Internet, aunque es poco intuitiva lo que implica una curva de aprendizaje muy grande.

Enterprise Architect: es una herramienta fácil de usar, entender y manejar, y generar gráficos de calidad. Es una herramienta propietaria, aunque existe una licencia freeware que brinda funcionalidades importantes pero es bastante limitada y no permite llevar a cabo todas las tareas que se necesitan.

JUDE: esta herramienta es personalizable que genera la documentación de los objetos de los diagramas con sus respectivos atributos y características y al igual que Enterprise Architect tiene licencia propietaria pero cuenta con una distribución Freeware JUDE Community, la cual está limitada, y no brinda la funcionalidad que se requiere para el proyecto.

CONCLUSIONES

Para el desarrollo del estudio comparativo, fue necesario realizar una búsqueda a profundidad, sobre cada una de las herramientas seleccionadas con el fin de abstraer las características principales de las mismas, y así elegir los parámetros adecuados para llevar a cabo la comparación deseada.

BIBLIOGRAFÍA

GARCÍA Félix, RUÍZ Francisco, CRUZ José Antonio, PIATTINI Mario. Gestión Integrada del Modelado y de la Medición del proceso de Software. Ciudad Real, España : s.n.

GARCÍA, Iván, GÓMEZ Jorge y PAVÓN Juan. Representación de las Relaciones en los Metamodelos con el Lenguaje Ecore. Madrid, España : s.n.

KELLY, Steven. GOPRR Description, 1997.

En la búsqueda de las herramientas se pudo corroborar que existe un amplio número de opciones de tipo libre, que brindan funcionalidades importantes, para lograr llevar a cabo proyectos de desarrollo de software de alta calidad y a bajo costo, que están a la altura de las de tipo propietario ofrecidas en el mercado.

Para realizar la comparación, se consideraron ocho características principales, a cada una de las cuales se les asignó un peso de acuerdo a la relevancia que representaban para el desarrollo de los proyectos. Los parámetros que obtuvieron un mayor valor, fueron los correspondientes a “soporte a UML” y “Soporte al estándar XMI”, debido a que éstos permiten lograr la interoperabilidad entre sistemas, facilitando la utilización de otras herramientas y garantizando su correcto funcionamiento.

De igual manera, las características de plataforma y estabilidad de la herramienta fueron consideradas entre las más relevantes debido a que estos criterios son definitivos para realizar una adecuada selección, dependiendo de la naturaleza del proyecto y de las demás aplicaciones que se deban utilizar.

Como complemento del estudio comparativo realizado, se está trabajando en la evaluación de herramientas de motores workflow, así como las que tienen que ver con el soporte a la gestión de proyectos, con el fin de establecer las más adecuadas para utilizarlas en el desarrollo de los proyectos planteados por cada uno de los autores.

INFOGRAFÍA

Baeza, Óscar J. 2005. MonoUML, deja que el mono trabaje por tí. [En línea] 30 de 03 de 2005. <http://www.genbeta.com/linux/monouml-deja-que-el-mono-trabaje-por-ti>.

Borland®. 2009. Borland® Together®. [En línea] 2009. [Citado el: 01 de Julio de 2009.] <http://www.borland.com/us/products/together/index.html>.

BOUML. 2009. BOUML. [En línea] <http://bouml.free.fr/index.html>, 2009.
Budinsky, Frank, 2006. 2006.

Capítulo de Estudiantes de ACM. 2003. Día. [En línea], 2003.
<http://acm.escet.urjc.es/documentos/acmlux/html/x776.html>.

change-vision. 2006. JUDE Community. [En línea] 26 de 02 de 2006. <http://jude.change-vision.com/jude-web/product/index.html>.

Corel Corporation. 2009. iGrafx. [En línea] 2009. [Citado el: 05 de Julio de 2009.] <http://www.corel.com/servlet/Satellite/us/en/Product/1152105061837#tabview=tab0>.

Favre, Liliana. 2005. Arquitectura de software Dirigida por Modelos MDA. Buena Aires, Argentina : s.n., 2005.

Fujaba. 2007. Fujaba Tool Suite 5. [En línea] 03 de 07 de 2007. <http://www.fujaba.de/>.

Gago, Sergio. 2006. Herramientas de trabajo III - UML - Poseidón. [En línea] 18 de Diciembre de 2006. [Citado el: 16 de Julio de 2009.] <http://sergiogh.aresdesarrollo.es/2006/12/18/herramientas-de-trabajo-iii-uml-poseidon/>.

IBM. 2009. Rational Rose Enterprise. [En línea] 2009. [Citado el: 05 de Julio de 2009.] http://www-142.ibm.com/software/dre/ecatalog/detail.wss?locale=es_ES&synkey=M221280M46834Z27.

Larrucea, Xabier, y otros. 2007. Especificación del metamodelo a utilizar. [En línea] 22 de Noviembre de 2007. [Citado el: 06 de Junio de 2009.] www.ines.org.es/.../d2-1-especificacion-del-metamodelo-spem-2-0-v10.doc.

Magic Draw. 2009. MagicDraw. [En línea] 2009. [Citado el: 04 de Julio de 2009.] <http://www.magicdraw.com/>.

Microsoft Corporation. 2009. Microsoft Office Visio 2007. [En línea] 2009. [Citado el: 05 de Julio de 2009.] <http://office.microsoft.com/es-es/visio/FX100487863082.aspx>.

Montlva, Jonás A. 2006. Modelado de procesos de Software. [En línea] 24 de 10 de 2006. [Citado el: 21 de 03 de 2009.] <http://www.vaneduc.edu.ar/Uai/facultad/sistemas/informes/Conferencia%20Modelado%20de%20Procesos%20SW.pdf>.

Navarro, Juanjo. 2005. UMLet 1.5. [En línea] 26 de 05 de 2005. <http://www.versioncero.com/noticia/19/umlet-15>.
NetBeans. NetBeans IDE 6.5 Features. [En línea] <http://www.netbeans.org/features/uml/>.

Odutola, Kunle y van der Wulp, Michiel. 2008. ArgoUML Quick Guide. [En línea] 2008. [Citado el: 30 de 05 de 2009.] <http://argouml-stats.tigris.org/documentation/quick-guide-0.28/>.

Omondo. 2009. Omondo. [En línea] 2009. <http://www.omondo.com/>.

Open Model Sphere. 2009. Open Model Sphere. [En línea] 2009. <http://www.modelsphere.org/use.html>.

Oracle. Oracle JDeveloper 11g. [En línea] http://www.oracle.com/tools/jdev_home.html.

Papyrus. 2008. Papyrus. [En línea] 14 de 03 de 2008. <http://www.papyrusuml.org/scripts/home/publigen/content/templatelates/show.asp?L=EN&P=55&vTicker=alleza&ITEMID=3>.

Pavlich, Jaime. 2005. De Modelos, Metamodelos y Metametamodelos . [En línea] 13 de 10 de 2005. [Citado el: 15 de 04 de 2009.] HYPERLINK "<http://tasof-ucn.blogspot.com/2005/10/de-modelos-metamodelos-y.html>" <http://tasof-ucn.blogspot.com/2005/10/de-modelos-metamodelos-y.html>.

Power Designer 2008. Power Designer. [En línea] 2008. [Citado el: 10 de Julio de 2009.] <http://www.mtbase.com/productos/modelamientometadatos/powerdesigner>.

Rational . 1998. Rational Unified Process Best Practices for Software Development Teams. 1998.

Skinner, Grant. 2003. GMODELER. [En línea] 2003. <http://www.gskinner.com/gmodeler/>.

Sparx Systems Pty Ltd. 2008. ENTERPRISE ARCHITECT. [En línea] 2008. [Citado el: 28 de Junio de 2009.] <http://www.sparxsystems.es/>.

StarUML. StarUML - The Open Source UML/MDA Platform. [En línea] <http://staruml.sourceforge.net/en/about.php>.

TCM. 2003. Toolkit for Conceptual Modeling (TCM). [En línea] 20 de 01 de 2003. <http://wwwhome.cs.utwente.nl/~tcm/>.

Umbrello. 2008. Umbrello UML Modeller. [En línea] 08 de 02 de 2008. <http://uml.sourceforge.net/>.

Visual Paradigm. Visual Paradigm for UML. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.