



# V Jornadas de Trabajo DYNAMICA

(DYNAmic and Aspect-Oriented  
Modeling for Integrated Component-  
based Architectures)

En Valencia, 23 y 24 de noviembre de 2006

Organización:



Colaboradores:



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



DEPARTAMENTO DE SISTEMAS  
INFORMÁTICOS Y COMPUTACION

# ACTAS

## V Jornadas de Trabajo DYNAMICA

(DYNAmic and Aspect-Oriented Modeling for  
Integrated Component-based Architectures)

En Valencia, 23 y 24 de noviembre de 2006

### Editores:

Jennifer Pérez  
Manuel Llavador  
Cristóbal Costa  
Nour Ali



Financiado por el CICYT (Centro de Investigación Científica Y Tecnológica), Proyecto DYNAMICA (DYNAmic and Aspect-Oriented Modeling for Integrated Component-based Architectures)

## **Editores**

### **Jennifer Pérez**

Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
Camino de Vera s/n  
E-46022 Valencia, Spain  
E-mail: [jeperez@dsic.upv.es](mailto:jeperez@dsic.upv.es)

### **Manuel Llavador**

Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
Camino de Vera s/n  
E-46022 Valencia, Spain  
E-mail: [mllavador@dsic.upv.es](mailto:mllavador@dsic.upv.es)

### **Cristóbal Costa**

Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
Camino de Vera s/n  
E-46022 Valencia, Spain  
E-mail: [ccosta@dsic.upv.es](mailto:ccosta@dsic.upv.es)

### **Nour Ali**

Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
Camino de Vera s/n  
E-46022 Valencia, Spain  
E-mail: [nourali@dsic.upv.es](mailto:nourali@dsic.upv.es)

©Los autores

Distribuido en España, Grupo de Ingeniería del Software y Sistemas de Información, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia  
Noviembre, 2006

## **Presidente de las jornadas**

Isidro Ramos Salavert

## **Comité Organizador**

**Presidente:** Jennifer Pérez Benedí

**Vocales:** Nour Ali Irshaid  
Silvia Mara Abrahao  
M<sup>a</sup> Eugenia Cabello Espinosa  
Jose Hilario Canós Cerdá  
Jose Ángel Carsí Cubel  
Cristóbal Costa Soria  
Abel Gómez Llana  
Emilio Insfrán Pelozo  
Patricio Letelier Torres  
Rogelio Limón Cordero  
Manuel Llavador Campos  
Elena M<sup>a</sup> Navarro Martínez  
M<sup>a</sup> Carmen Penadés Gramaje  
Carlos Solís Pineda

## **Comité Técnico**

Isidro Ramos Salavert, *Universidad Politécnica de Valencia*

Bárbara Álvarez Torres, *Universidad Politécnica de Cartagena*

Coral Calero Muñoz, *Universidad de Castilla-La Mancha*

Francisco José Rodríguez Urbano, *Universidad Carlos III*

Ambrosio Toval Álvarez, *Universidad de Murcia*

## **Subproyectos y Grupos Participantes**

### **SUBPROYECTO: PRISMA – Plataforma OASIS para Modelos Arquitectónicos**

TIC2003-07804-C05-01

Grupo de Investigación "Ingeniería de Software y Sistemas de Información" (ISSI)

Departamento de Sistemas Informáticos y Computación (DSIC)

Universidad Politécnica de Valencia (UPV)

### **SUBPROYECTO: ANCLA – Arquitecturas DiNámiCas para Sistemas de TeLeoperAción**

TIC2003-07804-C05-02

Grupo DSIE (División de Sistemas e Ingeniería Electrónica)

Departamento de Tecnologías de la Información y Comunicaciones

Universidad Politécnica de Cartagena (UPCT)

### **SUBPROYECTO: CALIPO – CALIdad en PORTales**

TIC2003-07804-C05-03

Grupo ALARCOS

Departamento de Informática

Universidad de Castilla-La Mancha (UCLM)

### **SUBPROYECTO: CARATE – Controlador de Arquitectura Reconfigurable para aplicaciones de Teleoperación**

TIC 2003-07804-C05-04

Grupo Laboratorio de Sistemas Inteligentes (LSI)

Departamento de Ingeniería de Sistemas y Automática (ISA)

Universidad Carlos III (UC3M)

### **SUBPROYECTO: PRESSURE – PREcise Software modeIS and reqUirements REUse**

TIC 2003-07804-C05-05

Grupo de Investigación en Ingeniería del Software (GIS)

Departamento de Informática y Sistemas

Universidad de Murcia (UMU)

# Presentación

El Proyecto DYNAMICA (**DYN**amic and **A**spect-Oriented **M**odeling for **I**ntegrated **C**omponent-based **A**rchitectures) ha sido un proyecto coordinado de tres años de duración financiado por el Ministerio de Ciencia y Tecnología. DYNAMICA surgió por los intereses comunes de cinco grupos de investigación españoles: el grupo de Ingeniería del Software y Sistemas de Información (ISSI) de la Universidad Politécnica de Valencia (UPV), el grupo de División de Sistemas e Ingeniería Electrónica (DSIE) de la Universidad Politécnica de Cartagena (UPCT), el grupo ALARCOS de la Universidad de Castilla-La Mancha (UCLM), el grupo del Departamento de Ingeniería de Sistemas y Automática (ISA) de la Universidad Carlos III (UC3M) y el Grupo de Investigación en Ingeniería del Software (GIS) de la Universidad de Murcia (UMU).

Las últimas jornadas DYNAMICA nos proveen un espacio para compartir los resultados obtenidos a lo largo del proyecto. Así, el objetivo de estas jornadas es presentar los resultados en forma de comunicaciones y “demos” orientadas a enfrentar la dinámica del software desde una visión arquitectónica en dominios con problemas reales como los sistemas de teleoperación, los sistemas hidráulicos y los portales Web.

Se han recibido 16 contribuciones en forma de 11 comunicaciones y 5 “demos” desde los diferentes nodos, reflejando el espíritu e ilusión que hemos invertido en este proyecto. Además, las relaciones inter-nodos han sido intensas, mostrando que somos conscientes que es el único modo de conseguir un avance hacia un objetivo común es mediante el trabajo colectivo. La amistad y sinergia entre todos nosotros ha permitido un excelente trabajo y un gran resultado de este proyecto.

Esto nos ha situado en primera línea de investigación en Ingeniería del Software tanto nacional como internacionalmente y ha propiciado el nacimiento de META, el nuevo proyecto en el que compartiremos los esfuerzos gran parte de los miembros del consorcio.

La voluntad de estar juntos, de aceptarnos mutuamente, con las discusiones necesarias, han sido el acicate dialéctico de avance de DYNAMICA que animamos a continuar en META.

Isidro Ramos.

Presidente de las Jornadas

# Índice

## Artículos

<i>Towards a Data Quality Framework for Web Portals</i> Angélica Caro, Coral Calero.....	13
<i>PtSM: Modelo de Selección de Portlets</i> M <sup>a</sup> Ángeles Moraga, Coral Calero, Mario Piattini .....	23
<i>Proyecto ANCLA: resumen de aportaciones, resultados y conclusiones</i> Grupo DSIE.....	33
<i>Perspectivas de futuro del desarrollo basado en modelos</i> Diego Alonso Cáceres, Bárbara Álvarez Torres y Pedro Sánchez Palma.....	49
<i>Verificación de Propiedades de Dominio mediante Modelado Preciso</i> Francisco Javier Lucas Martínez, Ambrosio Toval Álvarez, Francisco J. Ortiz.....	59
<i>Selección de Componentes basada en Requisitos en el Marco de SIREN</i> Miguel A. Martínez, Manuel F. Bertoa, Antonio Vallecillo, Ambrosio Toval.....	71
<i>Esquemas de control PID para una maqueta hidráulica</i> Francisco Rodríguez, Felipe Zottola, Sandra Alonso, Lucía Pintos.....	83
<i>Ambient-PRISMA: Ambients in Distributed and Mobile Aspect-Oriented Software Architectures</i> Nour Ali e Isidro Ramos.....	93
<i>Coordinación y Acceso a Información en Gestión de Emergencias</i> José H. Canós, Manuel Llavador, Carlos Solís, Patricio Letelier, M <sup>a</sup> . Carmen Penadés, Marcos R. S. Borges .....	103
<i>Hacia la Construcción de Arquitecturas Software Dinámicas</i> Cristóbal Costa, Jennifer Pérez, José Ángel Carsí.....	109
<i>Introducción al Cálculo-<math>\pi</math> con Prioridad</i> Carlos E. Cuesta, Jennifer Pérez, M. Pilar Romay .....	121

## **Demostraciones**

### *PRISMA CASE*

Jennifer Pérez, Ismael Carrascosa, Javier Guillén, José A. Carsí, Isidro Ramos.....139

### *MOMENT: Una herramienta de Gestión de Modelos aplicada a la Ingeniería Dirigida por Modelos*

Abel Gómez, Artur Boronat, Pascual Queralt, José Á. Carsí, Isidro Ramos.....141

### *LUNA+: Un entorno para la validación de modelos basado en prototipado automático*

Ángel Roche y Patricio Letelier.....143

### *Una herramienta visual para la generación automática de entornos de desarrollo software basados en modelos*

Manuel Llavador y José H. Canós .....145

### *MDHDM: Un Método de Desarrollo Hipermedia Dirigido por Modelos*

Carlos Solís, José H. Canós, María C. Penadés .....147

## PtSM: Modelo de Selección de Portlets

M<sup>a</sup> Ángeles Moraga, Coral Calero, Mario Piattini

Grupo de Investigación ALARCOS.  
Instituto de Desarrollo e Investigación UCLM-SOLUZIONA.  
Universidad de Castilla-La Mancha  
Paseo de la Universidad, 4. 13071. Ciudad Real, España  
{mariaangeles.moraga, coral.calero, mario.piattini}@uclm.es

**Abstract.** El uso de los portales Web continúa aumentando, lo que muestra su importancia dentro de la sociedad de la información actual. Hoy en día, los usuarios pueden cambiar de forma fácil de un portal a otro, por lo que si no queremos que nuestro portal desaparezca debemos mejorar/calcular la calidad del mismo. En concreto, en este artículo nos centramos en un tipo especial de portales que son los portales construidos mediante la agregación de portlets. Los portlets son componentes web y pueden ser pensados como COTS pero en un escenario Web. Por tanto, los desarrolladores de los portales tendrán que seleccionar el portlet más adecuado para su portal de entre un conjunto de portlets que tienen un conjunto parecido de funciones para las tareas y los objetivos específicos de los usuarios. Este artículo presenta un modelo de selección de portlets que guía al desarrollador del portal a realizar dicha selección.

**Keywords:** modelo de calidad, medidas, portal web, portlets

### 1 Introducción

Los portales web son aplicaciones basadas en Internet que permiten acceso a diferentes fuentes (proveedores) a través de una única interfaz [10] que proporciona personalización, facilidades en el registro (los usuarios sólo se tienen que registrar una vez) y agregación de contenido desde diferentes fuentes [8].

Pero las características de los portales pueden diferir de unos a otros, obteniendo así diferentes tipos de portales. Actualmente, a pesar del reconocimiento por parte de los investigadores de la existencia de diferentes tipos de portales, no se ha llegado a un consenso en cuanto a la clasificación de los mismos.

De entre todas las posibles clasificaciones destacamos la realizada en [1], donde dividen a los portales en generaciones en función de la evolución que han sufrido, obteniendo así la siguiente clasificación:

- *Portales de primera generación:* estos portales solían presentar una arquitectura software monolítica que se encargaba del desarrollo y mantenimiento del portal.
- *Portales de segunda generación:* permiten que los usuarios se creen una o más páginas personales compuestas de *portlets* personalizables, es decir,

miniaplicaciones web, locales o remotas, que representan fragmentos de marcas (noticias, tiempo, deportes, etc.) que el portal puede agregar en una página.

Sin embargo, los primeros portlets presentaban una falta de interoperabilidad entre ellos, como consecuencia de la inexistencia de un modelo común. Pero dicho problema fue resuelto tras la aparición, en el año 2003, del estándar WSRP (Web Services for Remote Portlets) [12]. Esto abre la posibilidad de la existencia de un mercado de portlets similar al mercado de componentes.

Por tanto, si se quiere un “buen” portal será necesario seleccionar los portlets más adecuados para construirlos. Por ello, será necesaria la utilización de modelos de calidad que ayuden a los desarrolladores de los portales a seleccionar el portlet más adecuado.

A pesar de que existen modelos de calidad para diferentes contextos, un modelo de calidad específico para la selección de portlets no existe. Por este motivo, nuestro objetivo consiste en desarrollar un modelo de selección de portlet (PtSM) que permita determinar cuál es el mejor portlet de entre un conjunto de portlets que tienen un conjunto parecido de funciones para las tareas y los objetivos específicos de los usuarios.

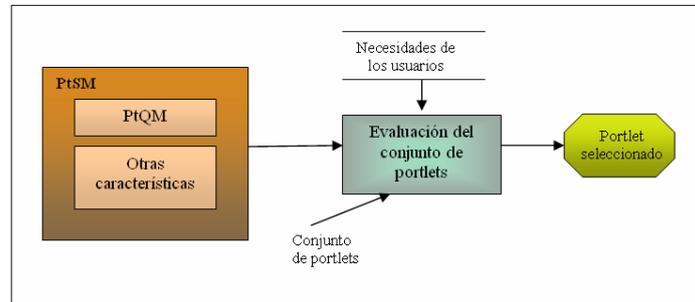
Este artículo se divide como sigue: en la sección segunda se presenta el modelo de selección de portlets, el cual está formado por un modelo de calidad de portlets que se presenta en la tercera sección y un conjunto de características que no están relacionadas con la calidad. Finalmente, en la sección cuarta se presentan algunas conclusiones y el trabajo futuro a realizar.

## 2 Modelo de selección de portlets

El modelo de selección de portlets, PtSM, permite a los desarrolladores de los portales, por un lado, determinar el mejor portlet de entre un conjunto de portlets que tienen un conjunto parecido de funciones para las tareas y los objetivos específicos de los usuarios, de acuerdo a su calidad y a otras características que no están relacionadas con la calidad pero que influyen en la selección. Y por otro lado, detectar los puntos débiles de cada portlet que hacen que su nivel de calidad disminuya – esta información puede ser utilizada posteriormente para incrementar la calidad de los mismos.

El uso del modelo de selección (PtSM) se resume en la figura 1. En primer lugar, se estiman los valores de las diferentes características que conforman el modelo para cada uno de los portlets evaluados. A continuación, considerando las necesidades de los usuarios se determina cuál es el mejor portlet de entre el conjunto de portlets evaluados. Obteniendo como resultado final el mejor portlet en función de cada caso.

Como se observa en la figura 1 PtSM está formado por un modelo de calidad de portlets (PtQM) y un conjunto de características que no están directamente relacionadas con la calidad.



**Fig. 1.** Uso del modelo de selección de portlets (PtSM)

Dentro del conjunto de características que no están directamente relacionadas con la calidad se pueden diferenciar, por ejemplo, aquellas que afectan al portlet como producto que tiene que ser comprado y agregado a nuestro software o la reputación o perfil del proveedor del portlet.

En concreto, para determinar el conjunto de características y subcaracterísticas que no están directamente relacionados con la calidad, se han analizado diferentes estándares. Estos estándares se pueden dividir en dos grupos: estándares dirigidos a herramientas CASE y estándares orientados a software de computación.

Por tanto, nuestro trabajo ha consistido en adaptar las características presentadas en estos estándares al contexto de los portlets. En la tabla 1, se presentan las características junto con sus subcaracterísticas y la fuente (o estándar) a partir del cual se ha adaptado la subcaracterística.

**Tabla 1.** Otras características no relacionadas con calidad, para la selección de portlets.

Característica	Subcaracterística	Fuente
Adquisición	Coste	ISO/IEC 14102
	Política de licencias	ISO/IEC 14102 e IEEE 1209
	Encuestas a grupos de usuarios	ISO/IEC 90003
Indicadores de soporte	Perfil del vendedor del portlet	ISO/IEC 14102 e IEEE 1209
	Perfil del portlet	ISO/IEC 14102 e IEEE 1209
	Información relativa a la instalación	ISO/IEC 90003
	Soporte del vendedor del portlet	IEEE 1209
	Realimentación del cliente para atender sus quejas	ISO/IEC 90003
Conformidad	Certificación del vendedor del portlet	ISO/IEC 14102 e IEEE 1209

### 3 Modelo de calidad de portlets (PtQM)

En este apartado se presentan las características de calidad que han sido tenidas en cuenta en PtSM y que se han agrupado en un modelo de calidad denominado PtQM.

Debido a que los portlets se pueden considerar como una mezcla entre los componentes software y las aplicaciones web, para la realización de este modelo se

V Jornadas de Trabajo Dynamica

han tenido en cuenta trabajos relacionados con productos software [7], con aplicaciones web [4, 9, 11, 13] y con componentes software [2, 3, 5, 14]. Cabe destacar que estos trabajos no pueden ser directamente aplicados al contexto de los portlets ya que éstos presentan algunas especificidades que los hacen diferentes, por lo tanto, sólo se han utilizado como base para la realización de nuestro modelo.

En las tablas 2 y 3 se muestran, respectivamente, las definiciones de las características y subcaracterísticas que finalmente han sido identificadas en PtQM.

**Tabla 2.** Definición de las características de calidad para el contexto de los portlets

Característica	Definición
Funcionalidad	Capacidad del portlet para proporcionar funciones que satisfagan las necesidades explícitas e implícitas cuando el portlet es utilizado en las condiciones especificadas.
Fiabilidad	Capacidad del portlet para mantener un nivel especificado de rendimiento cuando se utiliza en las condiciones especificadas.
Usabilidad	Capacidad del portlet para ser utilizado por los desarrolladores del portal cuando construyen el portal con él.
Eficiencia	Capacidad del portlet para proporcionar un rendimiento adecuado, relativo a la cantidad de recursos utilizados, bajo condiciones indicadas.
Reusabilidad	Capacidad del portlet para ser reusado en diferentes portales por diferentes desarrolladores.

**Tabla 3.** Definición de las subcaracterísticas de calidad para el contexto de los portlets.

Característica	Subcaracterística	Definición para el contexto de los portlets	
Funcionalidad	Exactitud	Capacidad del portlet para proporcionar los resultados o efectos correctos con el grado de precisión acordado.	
	Interoperabilidad	Interoperabilidad con el portal	Capacidad del portlet para interactuar con uno o más portales.
		Interoperabilidad con otros portlets	Capacidad del portlet para interactuar con otros portlets.
	Seguridad	Capacidad del portlet para prevenir accesos no autorizados, tanto accidentales como deliberados, a programas o datos.	
	Autosuficiencia	Capacidad del portlet para realizar por sí mismo la función que se espera que realice	
	Cohesión funcional	Capacidad del portlet para hacer uso de todos sus elementos a la hora de llevar a cabo su funcionalidad.	
	Conformidad	Capacidad del portlet para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.	

Fiabilidad	Madurez	Capacidad del portlet para evitar fallos provocados por errores en el software.
	Tolerancia a fallos	Capacidad del portlet para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.
	Recuperabilidad	Capacidad del portlet para recuperarse de fallos inesperados
	Degradabilidad	Esfuerzo necesario para re-establecer la funcionalidad esencial del portlet después de un fallo.
	Evaluabilidad	Capacidad del portlet para permitir a los desarrolladores del portal evaluar su forma y/o su contenido
	Disponibilidad	Capacidad del portlet para estar operativo todos los días del año 24/7/365
	Conformidad	Capacidad del portlet para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la fiabilidad.
Usabilidad	Comprensibilidad	Capacidad del portlet para permitir al usuario comprender cual es la funcionalidad del portlet.
	Facilidad de aprendizaje	Capacidad del portlet para permitir al usuario aprender como el portlet lleva a cabo sus objetivos
	Customizabilidad	Capacidad del portlet para ser customizado por los usuarios, consiguiendo reducir los esfuerzos necesarios para utilizarlo e incrementando la satisfacción del usuario con respecto al mismo.
	Conformidad	Capacidad del portlet para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la usabilidad.
Eficiencia	Comportamiento temporal	Capacidad del portlet para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo determinadas condiciones
	Utilización de recursos	Capacidad del portlet para utilizar cantidades y tipos de recursos apropiados cuando el portlet realiza su función bajo determinadas condiciones.
	Conformidad	Capacidad del portlet para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la eficiencia.
Reusabilidad	Comprensibilidad	Capacidad del portlet para permitir a los usuarios comprender cual es la funcionalidad del portlet.
	Portabilidad	Capacidad del portlet para ser transferido de un entorno a otro.

Una vez determinado el conjunto de características y subcaracterísticas que afectan a la calidad de los portlets, el siguiente paso consiste en definir atributos y medidas<sup>1</sup>

<sup>1</sup> Utilizamos la ontología de la medición del software (SMO), propuesta en [6], para presentar las medidas de una forma formal, utilizando conceptos y terminología obtenidos mediante consenso.

con el objetivo de poder determinar el nivel de calidad de cada una de las características.

Siguiendo la ontología de la medición propuesta en [6], cabe destacar que para el contexto que nos ocupa la clase de entidad es el “conjunto de portlets que proporcionan un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios”. Mientras que una entidad es “un portlet concreto al que se le aplica la medición”

La necesidad de información y los conceptos medibles serán diferentes para cada una de las características, por ejemplo, la necesidad de información para la usabilidad es “evaluar la usabilidad de un conjunto de portlets que tienen un conjunto parecido de funciones para las tareas y los objetivos específicos de los usuarios”, mientras que sus conceptos medibles son: comprensibilidad, facilidad de aprendizaje, customización y conformidad. Por tanto, los conceptos medibles de una determinada característica se corresponden con las subcaracterísticas de la misma.

En la tabla 4 se muestran, a modo de ejemplo los atributos y medidas identificados para el concepto medible comprensibilidad de la característica usabilidad.

**Tabla 4.** Otras características no relacionadas con calidad, para la selección de portlets.

Subcaracterística	Atributo	Medida			
		Medida Base	Escala	Tipo de escala	Unidad de medición
Comprensibilidad	Idioma del interfaz	Número de idiomas soportados por la interfaz	Número natural	Ratio	Lenguajes
	Documentación	El vendedor del portlet proporciona documentación	Boolean (0/1)	Nominal	Documentación
		Número de lenguajes en los que la documentación se proporciona	Número natural	Ratio	Lenguajes
		El portlet especifica su funcionalidad	Boolean (0/1)	Nominal	Descripción

Una vez identificados los atributos y las medidas, el siguiente paso consiste en calcular el indicador de cada una de las características identificadas. Un indicador es una medida derivada a partir de otras medidas que utiliza un modelo de análisis como forma de medir. El objetivo del modelo de análisis es definir un algoritmo para combinar una o más medidas con los criterios de decisión asociados. Estos criterios pueden ser valores umbrales, objetivos o patrones usados para determinar la necesidad de una acción o investigación posterior o para describir el nivel de confianza de un resultado dado.

Para calcular el indicador de cada característica, en primer lugar se tendrá que calcular un indicador para cada uno de los conceptos medibles asociados a la característica. A continuación, estos indicadores se transformarán (mediante la utilización de un modelo de análisis) para obtener el indicador de la característica. En la figura 2, se muestran a modo de ejemplo los diferentes indicadores que se tienen

que definir así como las transformaciones que se han de llevar a cabo para obtener el indicador de usabilidad.

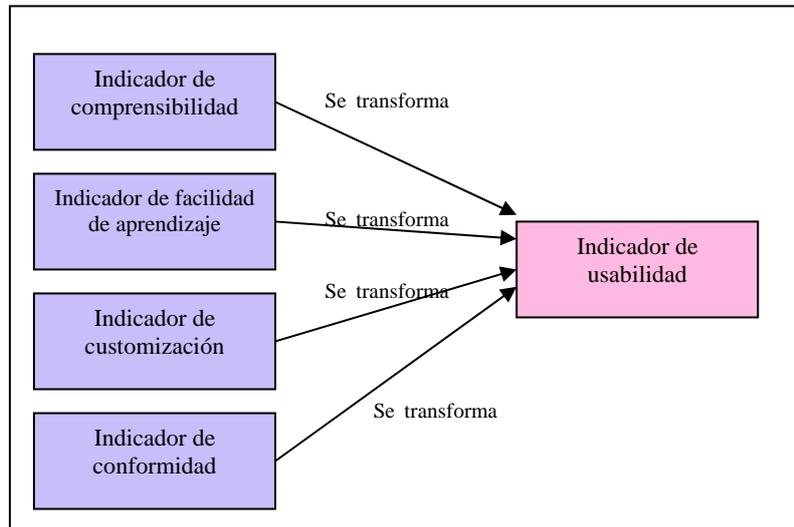


Fig. 2. Obtención del indicador de usabilidad.

Como se especificó anteriormente un indicador es una medida que es derivada de otras medidas utilizando un modelo de análisis como forma de medir. Por tanto, el primer paso a realizar consiste en definir un modelo de análisis. En este caso el modelo análisis es una función definida como sigue:

$$F(\text{Indicador}_x) = \text{Suma\_Rango\_Medidas} \quad (1)$$

Donde:

- $\text{Indicador}_x$  = representa a un indicador.
- $\text{Suma\_Rango\_Medidas}$  = representa la suma de los valores de los rangos<sup>2</sup> de las diferentes medidas definidas para el concepto medible que representa el indicador.

Para poder calcular el valor de la función F, en primer lugar, los valores de las medidas definidas para cada uno de los conceptos medibles se transforman en un rango de 0 a 5 y se indica cual es el valor mínimo aceptable que puede adquirir la medida y el valor máximo. Por ejemplo en la tabla 5 se muestra la transformación de los valores de la medida “número de idiomas soportados por la interfaz”, identificada para el concepto medible comprensibilidad de la característica funcionalidad, en un rango de 0 a 5.

<sup>2</sup> Los valores de las medidas se transforman en un rango que puede oscilar entre 0 y 5, con ello se consigue que todas las medidas adquieran el mismo conjunto de valores y por tanto, se facilita el análisis de los resultados de la función.

V Jornadas de Trabajo Dynamica

**Tabla 5.** Rango para la medida “número de idiomas soportados por la interfaz”

Rango	Valor de la medida
0	1
1	1
2	1
3	1
4	2
5	> 2

Esto mismo se ha de realizar para cada una de las medidas identificadas. De forma que teniendo en cuenta los rangos definidos para cada medida, la función F para la comprensibilidad adquiere un valor máximo de 17.5 y un valor mínimo de 14.5. Por tanto, estableciendo intervalos equidistantes entre ambos valores, los niveles para el indicador de comprensibilidad quedan tal y como se muestran en la tabla 6.

**Tabla 6.** Niveles para el indicador de comprensibilidad

Indicador de comprensibilidad	Criterios de decisión
Excelente	$F=17.5$
Alta	$16.5 \leq F < 17.5$
Media	$15.5 \leq F < 16.5$
Aceptable	$14.5 \leq F < 15.5$
No aceptable	$F < 14.5$

Exactamente igual se ha de hacer para los indicadores de facilidad de aprendizaje, customización, y conformidad que se transformarán junto con el indicador de comprensibilidad en el indicador de usabilidad. Para realizar esta transformación se utiliza el modelo de análisis, es decir, la función F. Para ello los valores de los niveles que puede adquirir cada indicador son transformados en un rango de 0 a 5 de acuerdo a la tabla 7.

**Tabla 7.** Rango para los niveles que pueden adquirir los indicadores

Rango	Valor de la medida
0	No aceptable
1	Aceptable
2	Aceptable
3	Medio
4	Alto
5	Excelente

Finalmente, se calcula tanto el valor máximo como mínimo aceptable de la función  $F$ , obteniendo así el valor que la función  $F$  tiene que adquirir para que el nivel del indicador de usabilidad sea excelente (coincide con el valor máximo de la función) y no aceptable (cualquier valor inferior al valor mínimo aceptable). Los niveles intermedios se calculan estableciendo intervalos equidistantes entre el valor máximo y el valor mínimo aceptable. En la tabla 8 se muestran todos los niveles que puede adquirir el indicador de usabilidad.

**Tabla 8.** Niveles para el indicador de usabilidad

Indicador de usabilidad	Criterios de decisión
Excelente	$F=20$
Alta	$15.82 \leq F < 20$
Media	$11.66 \leq F < 15.82$
Aceptable	$7.5 \leq F < 11.66$
No aceptable	$F < 7.5$

Este mismo proceso se ha de seguir para calcular el indicador de funcionalidad, fiabilidad, eficiencia y reusabilidad.

## 4 Conclusiones y trabajo futuro

En este artículo se ha presentado un modelo de selección de portlets, denominado PtSM. Este modelo puede ser utilizado para seleccionar el portlet más apropiado de entre un conjunto de portlets que tienen un conjunto parecido de funciones para las tareas y los objetivos específicos de los usuarios. PtSM utiliza un modelo de calidad (PtQM) que integra y adapta, al caso de los portlets, diferentes modelos propuestos para aplicaciones Web, componentes Software y productos Software. Además tiene en cuenta otras características que aunque no son de calidad están relacionadas con el

hecho de que un portlet es un producto comercial y por tanto, son relevantes a la hora de seleccionar un portlet u otro.

Como trabajo futuro se tendrán que validar las diferentes medidas que han sido definidas así como los umbrales que han sido establecidos. Además, está prevista la aplicación del modelo a diferentes portlets reales que tienen parecida funcionalidad, con el objetivo de determinar que portlet es más adecuado para su integración en un determinado portal.

**Agradecimientos.** Este trabajo es parte del proyecto CALIPO (TIC 2003-07804-C05-03) y la red CALIPSO (TIN2005-24055-E) financiados por el Ministerio de Educación y Ciencia y el proyecto DIMENSIONS (PBC-05-012-1) financiado por FEDER y la Junta de Comunidades de Castilla-La Mancha.

## Referencias

1. Bellas, F., Standards for Second-Generation Portals. IEEE Internet Computing., 2004. 8(2). pp. 54-60.
2. Bertoa, M.F. and A. Vallecillo. Quality Attributes for COTS Components. 2002. 6th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2002). Málaga.pp. 54-66.
3. Botella, P., X. Burgués, J.P. Carvallo, X. Franch, J.A. Pastor, and C. Quer, Towards a Quality Model for the Selection of ERP Systems. Component-Based Software Quality, 2003. pp. 225-245.
4. Calero, C., J. Ruiz, and M. Piattini, Classifying web metrics using the web quality model. Online Information Review, 2005. 29(3). pp. 227-248.
5. Franch, X. and J.P. Carvallo, Using Quality Models in Software Package Selection. IEEE Software., 2003. 20(1). pp. 34-41.
6. García, F., M.F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, and M. Genero, Towards a consistent terminology for software measurement. Information and Software Technology, 2006. 48(8). pp. 631-644.
7. ISO/IEC, ISO/IEC 9126. Software Engineering-Product Quality. Partes 1 a 4. 2001. International Organization for Standardization/International Electrotechnical Commission.
8. Java Community Process. JSR 168 portlet specification. 2003.Disponible en: <http://www.jcp.org/en/jsr/detail?id=168>. Accedido: Enero.
9. Kalepu, S., S. Krishnaswamy, and S.W. Loke. Verity: A QoS Metric for Selecting Web Services and Providers. 2003. Fourth International Conference on Web InformationSystems Engineering Workshops (WISEW'03).pp. 131-139.
10. Mahdavi, M., J. Shepherd, and B. Benatallah. A Collaborative Approach for Caching Dynamic Data in Portal Applications. 2004. Proceedings of the fifteenth conference on Australian database Vol.27.pp. 181-188.
11. Moustakis, V., C. Litos, A. Dalivigas, and L. Tsironis. Website Quality Assessment Criteria. 2004. Ninth International Conference on Information Quality.pp. 59-73.
12. OASIS. Web Service for Remote Portals (WSRP). Version 1.0. 2003. Disponible en: <http://www.oasis-open.org/committees/wsrp/>. Accedido: Mayo 2006.
13. Offutt, A.J., Quality attributes of web software applications. IEEE Software., 2002. 19(2). pp. 25-32.
14. Simão, R.P. and A. Belchior. Quality Characteristics for Software Components: Hierarchy and Quality Guides. 2003. Component-Based Software Quality.pp. 184-206.