



**VII**  
**Jornadas sobre**  
**Innovación y**  
**Calidad del**  
**Software**

Universitat de les Illes Balears  
11 y 12 de Julio de 2002  
Palma de Mallorca

**LIBRO DE PONENCIAS**



**Universitat de les**  
**Illes Balears**

# LIBRO DE PONENCIAS

*Patrocinadores de las jornadas:*



GOVERN  
DE LES ILLES BALEARS  
Conselleria d'Innovació i Energia



**NTE**  
SA

inQA.labs  
Software Testing

Patronato EOI :



CECA  
Confederación Española  
de Cajas de Ahorros

ERICSSON

FUNDACION AUNA

FUNDACION VODAFONE

LA CAIXA

OFICINA ESPAÑOLA  
DE PATENTES Y MARCAS



# VII Jornadas sobre Innovación y Calidad del Software

Universitat de les Illes Balears  
11 y 12 de Julio de 2002  
Palma de Mallorca


## LIBRO DE PONENCIAS



Universitat de les  
Illes Balears



C/ Gregorio del Amo, 6  
Teléfs: 91 349 56 00/35  
Fax: 91 349 56 74  
28040 Madrid  
Dep. Legal: M-31076-2002  
Impresión: Artes Gráficas Mañas, S. L.  
[www.grupomanas.com](http://www.grupomanas.com)

 Esta publicación se ha realizado en papel reciclado

## INDICE

<b>Presentación</b> .....	5
<b>1. Mejora de los procesos de software en el marco del Capability Maturity Model (CMM)</b> Luisa Morales .....	9
<b>2. Mejora del Proceso Software: ¿Funciona?</b> José A. Calvo-Manzano, Gonzalo Cuevas, Tomás San Feliu, Antonio de Amescua, Magdalena Arcilla y Jose Antonio Cerrada .....	17
<b>3. “CMM puesto en práctica con... ¡Acción, Urgencia, Excelencia!”</b> Màrius A. Gómez y Josep Mir Casals .....	25
<b>4. PATTERNS, un enfoque práctico a la gestión del conocimiento en el área de gestión de procesos software</b> Elixabete Ostolaza, Nuria Quintano y Giuseppe Satriani .....	35
<b>5. Proyectos de gran tamaño: los límites del <i>Extreme Programming</i></b> Joan Bosch .....	43
<b>6. e-métricas: Métricas de Internet</b> Joaquín Ríos Boutín .....	53
<b>7. Plan de mejora de la calidad del software en Fagor Automation S.Coop</b> Goiuria Sagardui y Josu Onandia .....	63
<b>8. Validación Empírica de Métricas para Diagramas de Estados en UML</b> David Miranda, Marcela Genero y Mario Piattini .....	71
<b>9. Causas y consecuencias de la implantación de metodologías de desarrollo y medidas de la calidad del Software en organizaciones.p</b> <b>Un caso práctico</b> Juan Fco. Hdez. Ballesteros .....	81
<b>10. Las pruebas de rendimiento en el ciclo de vida del desarrollo de una aplicación basada en componentes distribuidos</b> Carlos Melero .....	89
<b>11. Desarrollo de una metodología para la gestión de proyectos informáticos según ISO 10.006 en las A.A.P.P.</b> R. Concepción Suárez, F. Ortega Fernández, V. Rodríguez Montequín y J. Villanueva Balsera.....	95
<b>12. Sistema de Estimación de Proyectos Software basado en Técnicas de Inteligencia Artificial</b> César Nistal y Miguel Angel Martínez .....	103
<b>13. Evaluación de proveedores mediante la gestión de un repositorio de proyectos con una herramienta de estimación</b> Ramiro Carballo Gutiérrez .....	113

## Presentación

Desde sus orígenes el Grupo de Calidad del Software de ATI se marcó como objetivo principal el fomento de la investigación y difusión de las materias relacionadas con la Calidad del Software, tratando de compaginar las vertientes académica y profesional que constituyen las dos caras imprescindibles de esta disciplina. Las VII Jornadas sobre Innovación y Calidad del Software constituyen un paso más de avance sobre las ediciones anteriores en el empeño de consolidar el prestigio y la consideración de nuestra asociación. Por segunda vez en su historia, la sede de las Jornadas se traslada fuera de Madrid, cuna inicial del grupo que acoge desde hace tiempo entre sus miembros a personas de toda la geografía española. En esta ocasión, el especial empeño mostrado por Antonia Mas y Esperança Amengual, miembros del Grupo de Calidad de Software en Baleares, ha dado como resultado esta séptima edición de las Jornadas. Una edición especial por el volumen de originales de calidad recibidos en el proceso de revisión (lamentablemente un 40% de las recibidas no ha podido ser admitida en las Jornadas por las limitaciones de tiempo existentes). También es la primera vez que se añade un seminario el día anterior a las jornadas siguiendo una línea que, desde el grupo, siempre se ha creído muy interesante.

En definitiva, esta edición de las Jornadas supone una nueva cita de un foro ya consolidado de debate y comunicación para quienes se interesan en la calidad y en la innovación en el desarrollo y la gestión del software. En la organización de las Jornadas se ha perseguido tanto la presencia de experiencias prácticas sobre calidad del software en el quehacer diario de los profesionales del área como la presentación de métodos novedosos de trabajo en el desarrollo y gestión de aplicaciones informáticas. Seguimos en la línea de llevar a este evento, además de los temas ya tratados tradicionalmente en las Jornadas (mejora de procesos, medición de software, aseguramiento y sistemas de calidad, certificación, etc.), algunos de los métodos y técnicas que supongan un alto grado de innovación en el desarrollo de software: *extreme programming*, patrones, etc.

Tras un proceso de evaluación rigurosa, la gran variedad de trabajos presentados como comunicaciones, son la expresión de las inquietudes y el trabajo desarrollado por los investigadores y profesionales de nuestra disciplina que han sido agrupados en torno a cuatro grandes áreas temáticas: mejora de procesos software, evaluación de la calidad de los sistemas, gestión en organizaciones de software e innovación en el desarrollo de aplicaciones. La calidad de los trabajos y la categoría de los ponentes son una garantía para el éxito de las presentes Jornadas y nos deben servir de aliento en la consolidación de la asociación y del grupo de trabajo como foro de debate y comunicación entre investigadores y profesionales del software.

La Universidad de les Illes Balears nos acogió desde el primer momento y apoyó con total interés la responsabilidad de celebrar en su recinto esta edición de las Jornadas. El resultado se aventura muy prometedor, y las VII Jornadas sobre Innovación y Calidad del Software, que se celebran son hoy una realidad. Es seguro que nos podremos ver de nuevo en una octava edición de las mismas.

No podemos más que desear que vuestra asistencia sea fructífera y agradecer la colaboración de todos aquellos, que aún sin estar presentes, han hecho posible este encuentro. Especialmente, queremos recordar que la organización de las Jornadas no habría sido posible sin el esfuerzo de todos los que aportaron su ayuda en las anteriores ediciones pues, no en vano, podemos decir que, como en la famosa cita de Newton, hemos subido a hombros de gigantes.

El Comité Organizador

## Comité Organizador

### Coordinadora de las VII Jornadas sobre Innovación y Calidad del Software

Antònia Mas Pichaco

#### Miembros del Comité

Idoia Alarcón Rodríguez  
*Grupo de Calidad de ATI Madrid*

Esperança Amengual Alcover  
*Grupo de Calidad de ATI Madrid*

Mercedes de la Cámara  
*Grupo de Calidad de ATI Madrid*

José Javier Dolado Cosin  
*Grupo de Calidad de ATI Madrid*

Luis Fernández Sanz  
*Grupo de Calidad de ATI Madrid (coordinador)*

Jordi Duatis Juárez  
*Grupo de Calidad de ATI Madrid*

Ana Rus Brox Alarcón  
*Grupo de Calidad de ATI Madrid*

Grupo de apoyo en la Universidad de les Illes Balears  
*Gabriel Fontanet Nadal*

Grupo de apoyo en la Universidad Europea CEES  
*Estrella Gómez Fernández*  
*David Atauri Mezquida*

## Comité de Revisión

### Coordinadora

Antònia Mas Pichaco  
*Universitat de les Illes Balears*

### Miembros del Comité

Esperança Amengual Alcover  
*Universitat de les Illes Balears*

Mercedes de la Cámara  
*Universidad Politécnica de Madrid*

José Javier Dolado Cosin  
*Universidad del País Vasco*

Jordi Duatis Juárez  
*NTE S.A.*

Luis Fernández Sanz  
*Universidad Europea CEES*

Nuria Juan Sierra  
*STL*

Antonia Mas Pichaco  
*Universitat des Illes Balears*

Ricard Duro Lago  
*Grupo de Métodos y Calidad del Software de ATI Barcelona*

Joaquín Ríos Boutin  
*Grupo de Métodos y Calidad del Software de ATI Barcelona*

Joan Tort  
*Grupo de Métodos y Calidad del Software de ATI Barcelona*

Edmundo Tovar  
*Universidad Politécnica de Madrid*



# Validación Empírica de Métricas para Diagramas de Estados en UML

David Miranda, Marcela Genero, Mario Piattini

Grupo de Investigación ALARCOS

Departamento de Informática

Universidad de Castilla - La Mancha

Paseo de la Universidad, 4

13071 - Ciudad Real - ESPAÑA

[dmiranda@proyectos.inf-cr.uclm.es](mailto:dmiranda@proyectos.inf-cr.uclm.es), {[marcela.genero](mailto:marcela.genero@uclm.es), [mario.piattini](mailto:mario.piattini@uclm.es)}@uclm.es

## RESUMEN

Persiguiendo el objetivo de producir Sistemas de Software Orientado a Objetos (SOO) de mejor calidad es necesario considerar la calidad desde las primeras etapas del ciclo de vida, prestando especial atención a la etapa de modelado conceptual. Utilizando el estándar de modelado, Unified Modelling Language (UML), en la etapa de modelado conceptual se realizan diagramas que cubren tanto aspectos estáticos como dinámicos. Si bien existen numerosos trabajos que abordan la calidad de los diagramas estáticos, como los diagramas de clases, no ocurre lo mismo con los diagramas dinámicos, como los diagramas de estados, actividad, etc. El objetivo del presente trabajo es presentar un conjunto de métricas para medir la complejidad de los diagramas de estados en UML y evaluar a través de un experimento controlado si tales métricas pueden ser usadas como indicadores tempranos de una subcaracterística de la mantenibilidad, la comprensibilidad.

## Palabras clave

Calidad del Software Orientado a Objetos (SOO), métricas, diagramas dinámicos en UML, diagramas de estados en UML, métricas de complejidad, validación empírica.

## 1 INTRODUCCIÓN

La calidad del software se ha convertido en nuestros días en un factor de discriminación entre productos, contribuyendo en gran medida al éxito de los mismos y al de la compañía que los fabrica.

En Ingeniería del Software se reconoce que el control de calidad del SOO debe garantizarse en las primeras etapas del ciclo de vida. De ahí que la calidad del modelado conceptual tenga una enorme relevancia en el software que finalmente se implementa. Utilizando UML [23] en la etapa de modelado conceptual se realizan diagramas que cubren tanto aspectos estáticos como dinámicos. Para evaluar la calidad de tales diagramas de manera objetiva es necesario contar con medidas cuantitativas que eviten sesgos en el proceso de evaluación.

Existen varios trabajos sobre métricas para medir la calidad de los diagramas estáticos como los diagramas de clases [8][11][17][18][22]. Sin embargo, apenas hay unas pocas referencias en la bibliografía sobre métricas para los diagramas dinámicos. Una de las primeras aproximaciones hacia la definición de métricas para diagramas dinámicos aparece en [13], donde se definen y aplican métricas para diagramas desarrollados con OMT (Object Modelling Technique) [28]. En [31] se plantean métricas dinámicas utilizadas para el modelado con Real-Time Object Oriented Modeling (ROOM) [30]. En [26] se definen medidas de complejidad para

modelos conceptuales orientados a objetos dirigidos por eventos. Así mismo, Brito e Abreu [7] y Poels [27] señalan que la definición de métricas para diagramas que capturen los aspectos dinámicos del SOO es un área relevante, un tanto descuidada en el ámbito de la medición del software. Este hecho nos motivó para definir métricas para los diagramas dinámicos, comenzando con los diagramas de estados en UML [19].

Si bien nuestro objetivo es centrarnos en una de las características más críticas de la calidad del SOO, la mantenibilidad [20], sabemos que ésta es un atributo externo de la calidad que solo puede ser evaluado una vez que el producto está terminado o próximo a su fin. Por ello nuestro objetivo es definir un conjunto de métricas para medir la complejidad (atributo interno de la calidad) de los diagramas de estados en UML y evaluar a través de experimentación si efectivamente existe correlación con una subcaracterística de la mantenibilidad, la comprensibilidad. Consideramos que evaluando (y si es necesario mejorando) la comprensibilidad de tales diagramas contribuiremos a facilitar la comprensibilidad del SOO que finalmente será implementado.

La definición de las métricas debe realizarse de forma metodológica y disciplinada, en pro de obtener métricas fiables. Por ello para la definición de las métricas que presentaremos en este artículo hemos tenido en cuenta algunas recomendaciones que aparecen en [6] y [10]. Hemos seguido un proceso que consta de tres etapas: definición de métricas, validación teórica y validación empírica.

Para la definición y validación teórica de las métricas [19], se siguió el marco DISTANCE propuesto en [27], lo que nos garantiza que las métricas propuestas miden exactamente lo que intentan medir (validez de constructo), y además que están en la escala de ratio, hecho fundamental a la hora de decidir el tipo de test estadístico que hay que aplicar para analizar los valores de las métricas. Pero no basta con

la validación teórica, la validación empírica es crítica para el éxito de cualquier actividad de medición [1][15][21][29] porque demuestra de forma empírica que las métricas propuestas sirven para el propósito para el que han sido definidas y que pueden ser útiles en la práctica. En este trabajo presentamos un experimento controlado realizado para obtener evidencia empírica de la relación entre las métricas de complejidad de los diagramas de estados en UML y la comprensibilidad de los mismos. En la sección 2 presentamos la definición de las métricas para la complejidad de los diagramas de estados en UML. En la sección 3 describimos el experimento que realizamos para validar empíricamente tales métricas. Por último en la sección 4 presentamos las conclusiones obtenidas y las líneas futuras de investigación.

## **2 UNA PROPUESTA DE MÉTRICAS PARA LA COMPLEJIDAD DE LOS DIAGRAMAS DE ESTADOS EN UML**

Siguiendo la idea de Fenton [14], que afirma que no es aconsejable definir una única medida que capture todos los aspectos de la complejidad, propusimos (ver tabla 1) un conjunto de métricas para medir la complejidad de los diagramas de estados en UML, teniendo en cuenta cada uno de los elementos o componentes de dichos diagramas.

Una propiedad importante de las métricas que definimos, de acuerdo con la recomendación que aparece en un trabajo sobre el futuro de las métricas para productos software [16], es que son métricas sencillas, fáciles de entender y de automatizar.

Finalmente otro aspecto que cabe destacar es que las métricas propuestas permitirán a los diseñadores que modelan SOO, elegir la mejor alternativa entre un conjunto de diagramas de estados en UML con contenido semántico equivalente.

Nombre de la métrica	Definición
NUMERO DE ACCIONES DE ENTRADA (NEntryA)	El número total de acciones de entrada, i.e., las acciones que se ejecutan cada vez que se entra en un estado.
NUMERO DE ACCIONES DE SALIDA (NExitA)	El número total de acciones de salida, i.e., las acciones que se ejecutan cada vez que se sale de un estado.
NUMERO DE ACTIVIDADES (NA)	El número total de actividades.
NUMERO DE ESTADOS INICIALES (NIS)	El número total de estados iniciales.
NUMERO DE ESTADOS FINALES (NFS)	El número total de estados finales.
NUMERO DE ESTADOS SENCILLOS (NSS)	El número total de estados sencillos, considerando también los subestados dentro de un estado compuesto.
NUMERO DE ESTADOS COMPUESTOS (NCS)	El número total de estados compuestos, i.e. aquellos que tienen estados con subestados anidados.
NUMERO DE TRANSICIONES (NT)	El número total de transiciones considerando sólo aquellas en la que el estado origen es distinto al estado destino.
NUMERO DE AUTOTRANSICIONES (NST)	El número total de transiciones donde el estado origen y el estado destino son el mismo.
NUMERO DE TRANSICIONES INTERNAS (NIT)	El número total de eventos que se tratan dentro de un estado sin abandonarlo.

Tabla 1. Métricas para la complejidad de los diagramas de estados en UML

### 3 UN EXPERIMENTO PARA VALIDAR EMPÍRICAMENTE LAS MÉTRICAS PROPUESTAS

En esta sección describimos un experimento que hemos realizado para validar empíricamente las métricas propuestas (ver tabla 1). Hemos seguido las recomendaciones que aparecen en [4], [25] y [32] sobre como preparar experimentos controlados. Para describir el experimento seguimos (con mínimos cambios) el formato proporcionado por Wohlin et al. [32].

El objetivo del experimento es comprobar si existe relación entre la complejidad de cada diagrama de estados, capturada por las métricas; y la comprensibilidad del mismo, que medimos a través del tiempo necesario para responder el cuestionario que lleva adjunto cada diagrama.

#### 3.1 Definición

Para la definición de los objetivos (ver tabla 2) utilizamos la plantilla GQM (Goal Question Metric) [2][3].

Analizar	<i>las métricas de complejidad para diagramas de estados en UML</i>
Con el propósito de	<i>Evaluar</i>
Con respecto a	<i>la capacidad de ser empleadas como indicadores de la comprensibilidad de los diagramas de estados en UML</i>
Desde el punto de vista de	<i>los diseñadores de SOO</i>
En el contexto de	<i>estudiantes en el último año de la Ingeniería Superior Informática y profesores del área de Ingeniería del Software en el Departamento de Informática en la Universidad de Castilla-La Mancha</i>

Tabla 2. Objetivo del experimento

### 3.2 Planificación

Antes de la definición del experimento tuvo lugar la planificación. La planificación dispone como será dirigido el experimento, incluyendo las siguientes actividades:

*Selección del contexto.* El contexto del experimento es un grupo relacionado con el área de Ingeniería del Software de la universidad, y por tanto el experimento se desarrolla en un ambiente no industrial. Los sujetos fueron 11 estudiantes y 8 profesores. Los estudiantes están matriculados en último curso de la Ingeniería Superior Informática en la Universidad de Castilla-La Mancha. Todos los profesores pertenecen al área de Ingeniería del Software.

El experimento es específico ya que se centró en métricas para capturar la complejidad de los diagramas de estados en UML. La posibilidad de generalizar a partir de este contexto específico será comentada más adelante cuando discutamos las amenazas a la validez externa del experimento. El experimento se refiere a un problema real, i.e., qué indicadores pueden utilizarse para evaluar la comprensibilidad de los diagramas de estados en UML.

*Selección de los sujetos.* Elegimos los sujetos por conveniencia, i.e. son estudiantes y profesores con suficiente experiencia en el diseño y desarrollo de SOO usando UML.

*Selección de las variables.* La variable independiente es la complejidad de los diagramas de estados en UML. La variable dependiente es la comprensibilidad de los diagramas de estados en UML.

*Instrumentación.* Los objetos usados en el experimento fueron 20 diagramas de estados. La variable independiente, la complejidad, fue medida a través de las métricas presentadas en la sección 2. La variable dependiente, la comprensibilidad, se evaluará a partir del

tiempo que cada sujeto emplea en contestar el cuestionario asociado a cada diagrama (al que denominamos tiempo de comprensibilidad). Suponemos que un diagrama de estados es más comprensible cuanto menor sea su tiempo de comprensibilidad.

*Formulación de hipótesis.* Con el experimento pretendemos evaluar las siguientes hipótesis:

- Hipótesis nula,  $H_0$ : No existe correlación significativa entre la complejidad de los diagramas de estados en UML y el tiempo de comprensibilidad.
- Hipótesis alternativa,  $H_1$ : Existe correlación significativa entre la complejidad de los diagramas de estados en UML y el tiempo de comprensibilidad.

*Diseño del experimento.* Elegimos un diseño intra-sujetos para el experimento, i.e., todos los sujetos tuvieron que resolver todas las tareas propuestas en todos los cuestionarios. Los cuestionarios se entregaron en distinto orden a cada sujeto y se recalzó que deberían de resolverlos según orden en el que se le entregaba.

### 3.3 Operación

En esta fase se recogen los resultados, e incluye las siguientes actividades:

*Preparación.* Cuando se hizo el experimento todos los estudiantes habían cursado 2 asignaturas de Ingeniería del Software, en las que aprendieron en profundidad cómo construir SOO con UML. Los profesores tienen como mínimo 3 años de experiencia en el diseño y desarrollo de SOO. Además todos los sujetos recibieron un sesión intensiva de entrenamiento, previa a la realización del experimento. Sin embargo ninguno conocía los aspectos que tratamos de estudiar, ni las hipótesis establecidas.

Preparamos el material que se entregó a los sujetos, consistente en un guía explicando la notación de los diagramas de estados en UML, y los 20 diagramas. Los diagramas se referían a diversos universos de discurso, pero eran lo suficientemente generales como para ser

fácilmente comprendidos por los sujetos. La complejidad de cada diagrama es diferente ya que, como se puede ver en la tabla 3, se trató de cubrir un amplio rango de los valores de las métricas.

Diagrama	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
NentryA	1	1	2	0	3	6	1	1	0	2	1	1	2	1	1	0	2	2	0	0
NexitA	1	0	0	0	2	6	0	0	1	1	2	1	1	1	0	0	0	0	1	0
NA	0	3	2	2	2	0	1	3	4	0	1	1	0	2	4	5	1	1	0	0
NIS	1	1	2	1	1	2	3	1	0	0	4	1	1	1	2	1	1	1	1	1
NFS	0	0	1	1	1	1	1	0	0	0	3	0	1	1	0	0	1	0	1	0
NSS	3	4	4	4	4	6	5	5	5	4	6	3	2	3	9	9	5	12	2	5
NCS	0	0	1	0	0	1	2	0	0	0	3	0	0	0	1	0	1	0	0	1
NT	5	7	7	9	10	13	10	11	9	5	16	5	3	6	12	21	8	24	5	11
NST	0	0	0	0	0	0	0	2	1	0	1	0	1	2	1	2	0	0	0	1
NIT	2	0	0	2	3	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0

Tabla 3. Valores de las métricas para cada diagrama de estados en UML

Cada diagrama contenía un cuestionario para evaluar si el sujeto realmente había comprendido el diagrama. Todos los cuestionarios constaban de cuatro preguntas conceptualmente iguales para cada diagrama y presentadas en el mismo orden. Cada sujeto tuvo que anotar la hora a la que comenzó a contestar el cuestionario y la hora en la que terminó. La diferencia entre ambas es lo que llamamos tiempo de comprensibilidad (expresado en segundos).

*Ejecución.* Entregamos a los sujetos el material descrito en el párrafo anterior y les explicamos la manera de proceder para realizar el experimento, i.e., cada sujeto tenía que resolver el cuestionario por sí mismo y disponía de tiempo ilimitado. Se indicó que disponían de un plazo de una semana, tras el cual deberían entregar los cuestionarios resueltos. Pasados los 7 días recogimos todos los cuestionarios con las respuestas a las preguntas y los tiempos de comprensibilidad de cada diagrama.

*Validación de datos.* Todos los cuestionarios se consideraron válidos porque todos los sujetos tenían suficiente experiencia en el tipo de tareas requeridas y habían respondido a todas las preguntas de los cuestionarios correctamente.

### 3.4 Análisis e Interpretación

Nuestro objetivo es averiguar si existe relación entre las métricas propuestas (ver sección 2) y el tiempo de comprensibilidad. Hemos

utilizado los datos recogidos en el experimento, para probar las hipótesis formuladas en la sección 3.2.

Primero, aplicamos el test de Kolmogorov-Smirnov para averiguar si los datos recogidos siguen una distribución normal. Como los datos no siguen la distribución normal decidimos realizar un test no paramétrico como el de la Rho de Spearman con un nivel de significación  $\alpha=0.05$ , lo que significa que tenemos un grado de confianza del 95% (i.e. la probabilidad de rechazar la hipótesis nula cuando ésta es falsa es del 95%, lo cual es estadísticamente aceptable).

Analizando la tabla 4, que muestra los coeficientes de correlación de Spearman entre las métricas y los tiempos de comprensibilidad, podemos afirmar que las únicas métricas que parecen tener correlación con el tiempo de comprensibilidad son, NA, NIS, NSS y NT. Podemos deducir esto debido a que sólo dichas métricas tienen una correlación cercana o mayor que 0.5, que es un valor umbral comúnmente aceptado para evaluar valores de correlación. Los coeficientes de correlación para el resto de métricas no se muestran en la tabla 4, debido a que su valor es bastante inferior a 0.5.

	NA	NIS	NSS	NT
<b>Tiempo de Comprensibilidad</b>			0.483	0.574
			0.500	0.581

Tabla 4. Coeficiente de Spearman entre los valores de las métricas y el tiempo de comprensibilidad

Si bien estos resultados son un tanto prematuros, muestran indicios que nos motivan para seguir experimentando con estas métricas. Sería importante replicar el experimento, tanto interna como externamente, en el futuro para comprobar si se confirman los indicios aparecidos. También sería necesaria la aplicación de las métricas con datos obtenidos en "proyectos reales".

### 3.5. Evaluación de la validez

Discutiremos varias cuestiones que amenazan a la validez del estudio empírico y el modo en el que intentamos evitarlas.

**Amenazas a la validez de la conclusión.** La validez de la conclusión trata sobre la validez estadística de las conclusiones. La única cuestión que puede afectar a la validez estadística de este estudio es el tamaño de la muestra de datos (380 valores, 20 diagramas y 19 sujetos), que quizás no sea suficientemente amplia para aplicar un test estadístico [5]. Somos conscientes de ello y consideramos los resultados del experimento como preliminares.

**Amenazas a la validez de constructo.** La validez de constructo es el grado de precisión

con el que la variables dependiente e independiente son medidas. Para medir la variable dependiente, la comprensibilidad consideramos una medida objetiva: el tiempo de comprensibilidad; que es el tiempo que cada individuo emplea en contestar las preguntas asociadas a cada diagrama y por tanto el tiempo necesario para comprenderlo. La validez de constructo de la variable independiente está garantizada por el marco de Poels y Dedene [26] [27] usado en la definición y validación teórica de las métricas [19].

**Amenazas a la validez interna.** La validez interna es el grado en el que se pueden obtener conclusiones sobre el efecto causal que la variable independiente tiene sobre la variable dependiente. Se contemplaron los siguientes aspectos:

*Diferencias entre sujetos.* Usando un diseño intra-sujetos, se redujo el error debido a la diferencia entre sujetos. Como señalan Briand et al. [4], en experimentos en Ingeniería del Software cuando tratamos con poblaciones pequeñas, las variaciones en la habilidad de los participantes son la mayor preocupación ya que es difícil de abordar por aleatorización o

bloqueo. En este experimento, estudiantes y profesores tenían similar grado de experiencia en el modelado con UML.

*Conocimiento del dominio de aplicación de los diagramas de estados en UML.* Los diagramas de estados procedían de diferentes universos de discurso, pero eran suficientemente sencillos de comprender por cada sujeto. De este modo, el conocimiento del dominio no afectó a la validez interna.

*Precisión en los valores del tiempo.* Los sujetos asumieron la responsabilidad de anotar la hora de comienzo y de finalización de cada cuestionario. Aunque somos conscientes que podrían introducir cierta imprecisión, preferimos confiar en ellos.

*Efecto del aprendizaje.* Los cuestionarios se entregaron en distinto orden a cada sujeto, para evitar los efectos del aprendizaje. Al entregar los cuestionarios se recalcó varias veces que era obligatorio resolverlos en el orden en que aparecían. Sin embargo, no controlamos si realmente siguieron el orden ya que cada sujeto realizó el experimento en solitario.

*Efectos de fatiga.* Como media cada sujeto tardó una hora en resolver todos los cuestionarios del experimento, por tanto el efecto de fatiga no es relevante. También el diferente orden de los cuestionarios ayudó a evitar este efecto.

*Efectos de persistencia.* Para evitar los efectos de persistencia, el experimento se realizó con sujetos que nunca habían hecho un experimento de este tipo.

*Motivación de los sujetos.* Los sujetos que realizaron el experimento han participado voluntariamente para ayudarnos en esta investigación.

*Otros factores.* Los efectos de plagio y la influencia entre sujetos realmente no se controlaron. Ahora bien, antes del inicio del

experimento se explicó a los participantes, entre otras cuestiones, que los cuestionarios los tendrían que resolver por sí mismos y que los resultados no nos servirían si comentaban entre ellos las respuestas durante la realización del experimento.

**Amenazas a la validez externa.** La validez externa es el grado en el que los resultados de la investigación pueden ser generalizados a la población estudiada y a otros ambientes de investigación. Cuanto más grande es la validez externa, más pueden generalizarse los resultados de un estudio empírico como una práctica real de Ingeniería del Software. Se han identificado dos amenazas a la validez externa que pueden limitar la posibilidad de generalizar los resultados:

*Materiales y tareas usadas.* Tanto los diagramas utilizados como las tareas requeridas son representativas de casos reales. No obstante, es necesario hacer más estudios empíricos tomando "casos reales" de compañías de software.

*Sujetos.* Para solucionar la dificultad de conseguir profesionales que quieran participar en el experimento, trabajamos con profesores y estudiantes avanzados de Ingeniería Superior Informática. De todos modos, somos conscientes de que se deben realizar más experimentos con profesionales para poder generalizar estos resultados. No obstante, en este caso, las tareas a desempeñar no requerían alto nivel de experiencia industrial, por tanto, realizar el experimento con estudiante y profesores puede resultar apropiado [1].

### 3.6 Presentación y Difusión

Como la difusión de los datos experimentales es importante para la replicación externa de los experimentos [9], el material de este experimento estará disponible en breve en la web <http://alarcos.inf-cr.uclm.es>.

## 4 CONCLUSIONES Y TRABAJO FUTURO

Dada la gran influencia que tiene la calidad del modelado conceptual en la calidad del SOO que finalmente se implementa es necesario evaluar la calidad de los diagramas que se producen en la etapa de modelado conceptual, i.e., al inicio del ciclo de vida. En este artículo nos hemos centrado en los diagramas de estado en UML, uno de los varios diagramas que permiten capturar los aspectos dinámicos de un SOO. Hemos presentado un conjunto de métricas para medir la complejidad de dichos diagramas, y a través de un experimento controlado hemos evaluado si verdaderamente tales métricas están relacionadas con un factor crítico de la calidad como lo es la comprensibilidad. Los resultados obtenidos, demuestran que sólo las métricas NA, NIS, NSS y NT parecen estar correlacionadas con el tiempo de comprensibilidad. Si bien los resultados parecen ser prometedores, deben considerarse como preliminares. Somos conscientes que para obtener resultados concluyentes es necesario realizar la replicación tanto interna como externa del experimento, como así también aplicar las métricas propuestas a "casos reales".

Una vez confirmada cuáles de las métricas son útiles en la práctica, nuestro objetivo es construir sistemas de predicción de la comprensibilidad basados en los valores de las métricas, utilizando técnicas avanzadas como los Árboles de Regresión Difusos [12] y los Prototipos-Deformables Borrosos [24].

Finalmente otra línea de trabajo de interés, es el estudio de la relación entre la complejidad de los diagramas de estados en UML y otras subcaracterísticas de la mantenibilidad, como la modificabilidad y la analizabilidad.

#### Agradecimientos

David Miranda disfruta de una beca FPI del Ministerio de Ciencia y Tecnología.

Esta investigación es parte del proyecto DOLMEN apoyado por CICYT (TIC 2000-1673-C06-06).

#### BIBLIOGRAFIA Y REFERENCIAS

- [1] V. Basili, F. Shull and F. Lanubille, "Building Knowledge through families of experiments", *IEEE Transactions on Software Engineering*, N<sup>o</sup> 4, July/August. 1999, pp. 456-473.
- [2] V. Basili and D. Weiss, "A Methodology for Collecting Valid Software Engineering Data", *IEEE Transactions on Software Engineering*, 10, 1984, pp. 728-738.
- [3] V. Basili and H. Rombach, "The TAME project: towards improvement-oriented software environments", *IEEE Transactions on Software Engineering*, 14(6), 1988, pp. 728-738.
- [4] L. Briand, S. Arisholm, F. Counsell, F. Houdek and P. Thévenod-Fosse, "Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions", *Empirical Software Engineering*, 2000.
- [5] L. Briand, S. Morasca, V. Basili, *Defining and Validating Measures for Object-based High-level Design*, ISERN-98-04, Fraunhofer International Software Engineering Research, Germany, 1998.
- [6] L. Briand, K. El Emam and S. Morasca. *Theoretical and empirical validation of software product measures. Technical report ISERN-95-03*, International Software Engineering Research Network, 1995.
- [7] F. Brito e Abreu, H. Zuse, H. Sahraoui, and W. Melo, "Quantitative Approaches in Object-Oriented Software Engineering". *ECOOP'99 Workshops, LNCS 1743*, A. Moreira and S. Demeyer (eds). Springer-Verlag. 326-337, 1999.
- [8] F. Brito e Abreu, and R. Carapuça, "Object-Oriented Software Engineering: Measuring and controlling the development process". *4th Int Conference on Software Quality, Mc Lean, Va, USA*, 1994.
- [9] A. Brooks, J. Daly, J. Miller, M. Roper and M. Wood. *Replication of experimental results in software*



- ingiering, *Technical report ISERN-96-10*, International Software Engineering Research Network, 1996.
- [10] C. Calero, M. Piattini, and M. Genero, "Empirical Validation of Referential Integrity", *Information and Software technology*, 2001.
- [11] S. Chidamber and C. Kemerer C, "A Metrics Suite for Object Oriented Design", *IEEE Transactions on Software Engineering*. 20(6), pp. 476-493, 1994.
- [12] M. Delgado, A. Gómez Skarmeta and L. Jiménez, "A regression methodology to induce a fuzzy model", *International Journal of Intelligent Systems*, 16, 2001.
- [13] K. Derr, *Applying OMT*, SIGS Books, Prentice Hall, 1995.
- [14] N. Fenton, "Software Measurement: A Necessary Scientific Basis", *IEEE Transactions on Software Engineering*, 20(3), pp. 199-206, 1994.
- [15] N. Fenton, and S. Pfleeger, *Software Metrics: A Rigorous Approach*, 2nd. Edition, Chapman & Hall, 1997.
- [16] N. Fenton and M. Neil, *Software Metrics: a Roadmap. Future of Software Engineering*, ACM, pp. 359-370, 2000.
- [17] M. Genero, J. Olivas, M. Piattini, and F. Romero, "Using metrics to predict OO information systems maintainability". *CAISE 2001*. Interlaken, Switzerland, LNCS 2068, 2001.
- [18] M. Genero, M. Piattini, and C. Calero, "Early Measures For UML class diagrams", *L'Objet*, 6(4), Hermes Science Publications, pp. 489-515, 2000.
- [19] M. Genero, D. Miranda and M. Piattini, "Defining and Validating Metrics for UML Statechart Diagrams". *6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2002)*, Málaga, 2002. (por aparecer)
- [20] *ISO 9126: Software Product Evaluation-Quality Characteristics and Guidelines for their Use*. ISO/IEC Standard 9126, Geneva. 1999.
- [21] B. Kitchenham, S. Pfleger, and N. Fenton, "Towards a Framework for Software Measurement Validation", *IEEE Transactions of Software Engineering*, 21(12), pp. 929-943, 1995.
- [22] M. Marchesi, "OOA Metrics for the Unified Modeling Language". *Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering*, pp. 67-73, 1998.
- [23] Object Management Group, *UML Revision Task Force, OMG Unified Modeling Language Specification*, v. 1.3, document ad/99-06-08, 1999.
- [24] J. A. Olivas and F. P. Romero, "FPKD. Fuzzy Prototypical Knowledge Discovery. Application to Forest Fire Prediction". *Proceedings of the SEKE'2000, Knowledge Systems Institute*, Chicago, Ill. USA, 2000.
- [25] D. Perry, A. Porter and L. Votta, "Empirical Studies of Software Engineering: A Roadmap. Future of Software Engineering", *Ed: Anthony Finkelstein, ACM*, pp. 345-355, 2000.
- [26] G. Poels, and G. Dedene, "Measures for Assessing Dynamic Complexity Aspects of Object-Oriented Conceptual Schemes". *19th International Conference on Conceptual Modelling (ER 2000)*. Salt Lake City, USA, pp. 499-512. 2000.
- [27] G. Poels, and G. Dedene, "Distance-Based software measurement: necessary and sufficient properties for software measures", *Information and Software Technology*, 42, pp. 35-46, 2000.
- [28] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, *Object-Oriented Modelling and Design*, Prentice Hall, USA, 1991.
- [29] N. Schneidewind, "Methodology For Validating Software Metrics", *IEEE*

- Transactions of Software Engineering*,  
18 (5), pp. 410-422, 1992.
- [30] B. Selic, G. Gullekson, and P. Ward,  
*Real-Time Object Oriented Modelling*,  
John Wiley & Sons, 1994.
- [31] S. Yacoub, H. Ammar and T. Robinson  
"Dynamic Metrics for Object Oriented  
Designs". *Proceedings of the Sixth*
- IEEE International Symposium on*  
*Software Metrics*, 1998.
- [32] C. Wohlin, P. Runeson, M. Höst, M.  
Ohlson, B. Regnell and A. Wesslén,  
*Experimentation in Software*  
*Engineering: An Introduction*, Kluwer  
Academic Publishers, 2000.