

THOMSON



X Jornadas de Enseñanza Universitaria de la Informática

JENUI 2004

ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE ALICANTE

ALICANTE DEL 14 AL 16 DE JULIO DE 2004



Universitat d'Alacant
Universidad de Alicante

X Jornadas de Enseñanza Universitaria de la Informática

THOMSON
★

Australia • Canada • Mexico • Singapore • Spain • United Kingdom • United States



Datos de catalogación bibliográfica

**X Jornadas de Enseñanza Universitaria de la Informática
Jenui 2004**

Thomson Paraninfo, S.A, Madrid 2004

ISBN: 84-9732-334-3

Materia: Informática 681.3
Enseñanza superior 378

Formato: 240x170

Páginas 546

© Los autores, 2004

Diseño de portada: Gabinete de Imagen y Comunicación Gráfica de la
Universidad de Alicante

Maquetación: El Comité Organizador

Primera edición: Julio de 2004

I.S.B.N: 84-9732-334-3

Depósito Legal: M. 31.179-2004

Imprime: Closas Orcoyen, S. L

Polígono Igarsa. Paracuellos de Jarama (Madrid)

**ACTAS DE LAS X JORNADAS DE ENSEÑANZA UNIVERSITARIA DE
LA INFORMÁTICA (JENUI 2004)**

ORGANIZADAS POR:

Grupo de Informática Industrial e Inteligencia Artificial (i³a)
Departamento de Ciencia de la Computación e Inteligencia Artificial
Escuela Politécnica Superior
Universidad de Alicante
Asociación de Enseñantes Universitarios de la Informática (AENUI)

ENTIDADES COLABORADORAS:

Vicerrectorado de Extensión Universitaria de la Universidad de Alicante
Conselleria d'Indústria, Comerç i Turisme, Generalitat Valenciana
Ministerio de Ciencia y Tecnología
Caja de Ahorros del Mediterráneo (CAM)
Novática
Thomson

COMITÉ DE PROGRAMA

PRESIDENTE

Cristóbal Pareja Flores	(Universidad Complutense de Madrid)
Marián Díaz Fondón	(Universidad de Oviedo)
Joaquín Ezpeleta Mateo	(Universidad de Zaragoza)
Jesús Joaquín García Molina	(Universidad de Murcia)
Rosalía Peña Ros	(Universidad de Alcalá)
Edmundo Tovar Caro	(Universidad Politécnica de Madrid)
Miguel Valero García	(Universitat Politècnica de Catalunya)

COMITÉ ORGANIZADOR
UNIVERSIDAD DE ALICANTE

PRESIDENTES DE HONOR

Salvador Ordóñez Delgado	(Secretario de Estado de Universidades e Investigación)
Miguel Louis Cereceda	(Rector de la Universidad de Alicante)

PRESIDENTES

Faraón Llorens Largo / Ramón Rizo Aldeguer

Pilar Arques Corrales
Fidel Aznar Gregori
Patricia Compañ Rosique
Fernando Llopis Pascual
Rafael Molina Carmona
Francisco Mora Lizán
Mar Pujol López
Rosana Satorre Cuerda
Pablo Suau Pérez
Carlos Villagrà Arnedo

COLABORADORES EN EL PROCESO DE REVISIÓN

José Joaquín Aguilera Garcia	UJAEN	Carlos López Nozal	UBU
Sergio Albiol	UNIZAR	Adolfo Lozano Tello	UNEX
Carlos Aliagas Castell	URV	Sergio Luján Mora	UA
Pedro J. Álvarez	UNIZAR	Javier Macías del Campo	UAH
Fernando Álvarez García	UNIOVI	Mercedes Marqués Andrés	UJI
Joan Aranda Lopez	UPC	Oiga Marroquín Alonso	UCM
Francisco Araque Cuenca	UGR	Antonio Martí	UPV
Luis Jesús Arévalo Rosado	UNEX	Raúl Marticorena Sánchez	UBU
Fidel Aznar Gregori	UA	Francisco Martínez Domínguez	UNIZAR
Patricia Balbastre Betoret	UPV	Margarita Martínez Santamarta	URJC
Sandra Baldassarri	UNIZAR	Raquel Martínez Unanue	URJC
José Ramón Balsas Almagro	UJAEN	Gloria Martínez Vidal	UJI
José Ángel Bañares Bañares	UNIZAR	Jorge Más Estellés	UPV
Pedro Blesa Pons	UPV	Manuel Mejías Risoto	US
José Manuel Burgos Ortiz	UPM	Marcos Menarguez Tortosa	UM
Antonio Cañas Vargas	UGR	Roc Messeguer Pellarés	UPC
Pedro José Clemente Martín	UNEX	Joe Miró Julià	UIB
Patricia Compañ Rosique	UA	Andrés Molina	UJAEN
Alberto de la Encina Vara	UCM	Rafael Molina Carmona	UA
Antonio J. de Vicente Rodríguez	UAH	M. Angels Moncusi	URV
Adelaida Delgado	UIB	Antonio Moreno Ribas	URV
Juan José Escribano Otero	UEM	Juan Manuel Murillo Rodríguez	UNEX
F. Javier Ferrández Pastor	UA	José Neira Parra	UNIZAR
José Flich Cardo	UPV	Carmen Nieves Ojeda Guerra	ULPGC
José Fortes Gálvez	ULPGC	Santiago Ortego Carazo	EUPMT
Isabel Gallego Fernández	UPC	María José Ortín Ibáñez	UM
Montse García	URV	Juan José Pantrigo Fernández	URJC
María José García Garia	UEM	Nieves Pavón Pulido	UHU
Pedro García López	URV	Pedro Luis Pérez Serrano	UNEX
Ginés García Mateos	UM	Angel F. Perles Ivars	UPV
Jesús García Molina	UM	Mario Piattini	UCLM
Javier García Zubía	DEUSTO	Juan Luis Posadas Yagüe	UPV
Piedad Garrido Picazo	UNIZAR	José Poveda	UJI
Daniel Gayo Avello	UNIOVI	José Luis Poza Luján	UPV
María José Gil Larrea	DEUSTO	Mar Pujol López	UA
Domingo Giménez Cánovas	UM	Jorge Ramio Aguirre	UPM
Estrella Gómez	UEM	Isabel Ramos	US
Mercedes Gómez Albarrán	UCM	Miguel Rebollo Pedruelo	UPV
Alberto Gómez Mancha	UNEX	Fernando Rubio Diez	UCM
Pilar González Ferez	UM	Francisco Ruiz González	UCLM
Julia González Rodríguez	UNEX	Fernando Sáenz Pérez	UCM
Antoni Grau	UPC	Julio Sahuquillo Borrás	UPV
Ángel Grediaga	UA	Fermin Sánchez Carracedo	UPC
Carlos Gregorio Rodríguez	UCM	Fernando Sánchez Figueroa	UNEX
José María Gutiérrez Martínez	UAH	Rosana Satorre Cuerda	UA
Diego Gutiérrez Pérez	UNIZAR	Clara María Segura Díaz	UCM

COLABORADORES EN EL PROCESO DE REVISIÓN

Iñaki Íñigo Ochoa de Chinchetru	UNIZAR	Diego Sevilla Ruiz	UM
Carmen Juan M.	UPV	Lourdes Tajés Martínez	UNIOVI
Vicente J. Julián Inglada	UPV	Silvia Terrasa	UPV
José Emilio Labra Gayo	UNIOVI	Jaime Urquiza Fuentes	URJC
María Jesús Lapeña Marcos	UNIZAR	Maite Urretavizcaya Loínaz	EHU
Pedro Latorre Andrés	UNIZAR	Miguel Ángel Vega Rodríguez	UNEX
Lenin G. Lemus Zúñiga	UPV	José Alberto Verdejo López	UCM
Patricio Letelier Torres	UPV	David Vernet Bellet	URL
Fernando Llopis Pascual	UA	María Teresa Villalba De Benito	UEM
Faraón Llorens Largo	UA	Ferrán Virgós	UPC
Natalia Lopez Barquilla	UCM	F. Javier Zarazaga Soria	UNIZAR
Pedro E. López De Teruel Alcolea	UM	Luisa Zúnica Ramajo	UPV

[...] Hay que renovar los métodos, haciéndolos activos, personales, tendiendo a favorecer la evolución formal del espíritu y hacer que éste busque y halle por sí mismo los materiales que han de construir su pensamiento y que sólo de esta suerte hace suyos [...]"

Francisco Giner de los Ríos

Escritos sobre la universidad española (1902)

Presentación

Introducción

Ésta es la X edición de las Jornadas de Enseñanza Universitaria de la Informática (JENÚI 2004), celebradas en Alicante los días 14, 15 y 16 de julio de 2004 en la Escuela Politécnica Superior de la Universidad de Alicante. La preocupación por mejorar los métodos, materiales y enfoques docentes no es nueva en la universidad, como refleja la fecha de la cita previa. Pero en el caso de la informática, cuyos contenidos, técnicas y métodos como disciplina están sufriendo una revolución permanente, los cambios en su docencia son sin duda más rápidos y profundos que en la mayoría de las disciplinas universitarias. Y sin embargo, en contraste con los centenares de congresos existentes sobre investigación en Informática, ésta es una de las poquísimas reuniones dedicadas a su docencia. Ahí reside la importancia de las JENUI, y por eso no extraña que, desde su nacimiento, estas jornadas hayan ido despertando cada vez mayor interés y se hayan ido consolidando; actualmente, podemos afirmar que la preocupación por la docencia de la informática ha dejado de ser una actividad menor en relación con la investigación.

Contenidos: líneas estratégicas, conferencias y otras actividades

En la presente edición se ha alcanzado un nivel de calidad del que nos sentimos orgullosos. Repasemos brevemente el contenido. Entre los cambios que más nos atañen actualmente en la docencia de la universidad, el proceso de integración en Europa es posiblemente el que más nos preocupa y el que va a tener repercusiones más profundas a corto y a medio plazo, tanto en los contenidos como en los métodos. Por esta razón, la primera línea estratégica es:

Adaptación al espacio europeo de educación superior: calidad de la docencia; adaptación al sistema de crédito europeo; métodos innovadores; nuevas exigencias al profesorado.

La otra línea estratégica subraya el principal objetivo de los futuros informáticos, que ha de dirigir nuestra labor docente: formarse para su profesión.

Formación para la profesión: adaptación a las necesidades de la empresa; colocación de los titulados; experiencias en el aprendizaje de habilidades profesionales.

A estos importantes temas están dedicados un 20% de las ponencias, entre otras actividades. Entre los contenidos de esta edición, deseamos destacar en primer lugar las conferencias del Prof. Bertrand Meyer (de enorme prestigio en el área de ingeniería del software orientado a objetos, creador del lenguaje Eiffel y la técnica de diseño por contrato que tanta influencia ha tenido en el mundo académico, es profesor del Instituto federal suizo -ETH, Eidgenössische Technische Hochschule- de Zurich), y la del profesor Josep Casanovas (decano de la Facultad de Informática de la Universidad Politécnica de Cataluña y coordinador del proyecto de la ANECA para la elaboración del Libro Blanco sobre las titulaciones de Ingeniería Informática en España) titulada *El Libro Blanco de la Ingenierías en Informática: el proyecto EICE*. Además, está programada una mesa redonda que debatirá sobre *La imagen académica, profesional y social del Ingeniero en Informática*.

Proceso de revisión

Una preocupación permanente es la calidad de las ponencias presentadas, y en ella el proceso de revisión desempeña un papel doblemente importante, por actuar como filtro y como orientación constructiva para mejorar distintos aspectos de los artículos. Con las pasadas ediciones, podemos considerar que este proceso está consolidado actualmente, por lo que se ha mantenido en lo esencial el procedimiento seguido en anteriores ediciones (doble anonimato, directrices, escalas de valoración, etc.), con sólo cambios mínimos: se ha aumentando la plantilla de revisores para que cada revisor no tenga que hacer en ningún caso más de tres revisiones; como contrapartida, se han podido elaborar informes más completos. Debemos agradecer el esfuerzo desinteresado de estos

expertos, que enriquecen a los autores y ayudan a mejorar la calidad de los trabajos. Como en las últimas ediciones, el trabajo de selección de las ponencias ha sido doblemente anónimo. Cada trabajo ha sido evaluado por 3 revisores al menos y, en casos de evaluaciones contradictorias, se ha solicitado una cuarta opinión. En los trabajos en que se ha considerado necesario incorporar alguna mejora a propuesta de los revisores, se han vuelto a revisar las versiones finales.

Estadística

El resumen estadístico de las contribuciones es el siguiente: se recibieron 118 trabajos, procedentes de 32 universidades. Teniendo en cuenta las evaluaciones, el Comité de Programa ha aceptado 62 trabajos, de los cuales 56 son ponencias y 6 son recursos docentes. Globalmente, estas cifras suponen una tasa de aceptación de un 53%, cifra algo inferior a la de la edición anterior.

Publicación

Como puede verse, estas actas se han publicado en la editorial Thomson, lo que da prestigio a las Jornadas y a dicha editorial y supone sin duda un aliciente para el público de las Jornadas. Queremos por ello agradecer a la editorial Thomson y en especial a D. Andrés Otero la confianza en nuestro trabajo. Por otra parte, la revista *Novática* (con una tirada de 6000 ejemplares y en cuya Web se visitan cada mes unas 100.000 páginas) publicará dos trabajos seleccionados entre los presentados, en su columna *enseñanza universitaria de la informática*. Debemos agradecer a Rafael Fernández Calvo esta colaboración y las facilidades que nos ha brindado. Estas colaboraciones suponen sin duda un estímulo para los participantes en las Jornadas y contribuyen a mejorar la calidad de los trabajos.

Sede

Este año tan especial del décimo aniversario de estas jornadas, se celebran en Alicante, una ciudad con tres mil años de antigüedad, levantada entre el mar y la montaña, que es la quinta provincia en la economía española, una de las de clima más benigno de Europa y de mayor atractivo, por sus hermosas playas, su cielo despejado y su geografía variada. Entre sus monumentos destacan el Castillo de Santa Bárbara, la iglesia de Santa María y el palacio de Gravina, así

como la isla de Tabarca, antiguo refugio de piratas. Su gastronomía es riquísima, compuesta por arroces como el negro, salazones como la mojama que ya elaboraban fenicios, griegos o cartagineses, por helados y turroneos sin igual en el mundo, y por los vinos rancios, generosos o mistelas; este lado de las jornadas merecería sin duda un capítulo aparte en caso de tener más espacio...

Pero Alicante es sobre todo una capital universitaria, con una intensa actividad cultural y una animación también intensa, que se extiende hasta el anochecer, sin duda favorecida por su clima. En estas líneas se ha de hacer constar la gratitud de todos nosotros al *Grupo de Informática Industrial e Inteligencia Artificial del Departamento de Ciencia de la Computación e Inteligencia Artificial* y a la *Escuela Politécnica Superior de la Universidad de Alicante*, que cumple este año sus veinticinco primaveras. Todos ellos nos acogen estos días con lo mejor de su cuidado, sus recursos y su hospitalidad.

Siguiente sede

La Asociación AENUI ha delegado en el Comité de Seguimiento la selección de la sede de las próximas Jornadas de JENUI. Finalmente, se ha designado la Universidad Europea de Madrid como sede para 2005. Desde ahora, animamos a todos a empezar a preparar nuestras contribuciones para el Jenui 2005. Los detalles irán publicándose paulatinamente; mientras tanto, consultad la siguiente dirección de Internet:

<http://jenui.uem.es>

Resumen

Estas actas pretenden ser una herramienta útil para quienes enseñamos informática en la universidad.

Y al trabajo

Damos la bienvenida a todos, deseando que sean unas jornadas agradables y provechosas, profesional y personalmente.

Alicante, 14 de julio de 2004
Comité de Programa y Comité Organizador

CONTENIDOS

Conferencias

Touch of class: teaching introductory programming outside-in 5
Bertrand Meyer

El Libro Blanco de la Ingeniería en Informática: el proyecto EICE 13
Josep Casanovas, José Manuel Colom, Iñaki Morlán, Ana Pont,
Maria Ribera Sancho

Ponencias

Tema estratégico: Adaptación al espacio europeo de educación superior ... 21

Un enfoque de informática de gestión para los estudios de Ingeniería
Informática en el marco de Bolonia 23
Francisco J. García Peñalvo, José A. Gómez Pires, Luis Alonso Romero,
Luis A. Martins do Amaral, José L. Pérez Iglesias
Universidad de Salamanca

Experiencia sobre el proyecto piloto de valoración de la Metodología para la
acreditación de las titulaciones propuesta por la ANECA 31
Agustín Cernuda del Río, Aquilino A. Juan Fuente, Marián Díaz Fondón,
David J. Brenes Martínez, Javier Otero Diz, Javier De Andrés Suárez,
Covadonga Nieto Fernández
Universidad de Oviedo

¿Están los alumnos preparados para el Tour de Francia? Comportamientos,
hábitos y Sistema de Créditos Europeo 39
Nieves Pavón Pulido
Universidad de Huelva

Convergencia hacia el Espacio Europeo de Educación Superior: Algunas ideas
prácticas y viables para llevar a cabo el cambio de paradigma 47
Marián Díaz Fondón, Miguel Riesco Albizu, Ana Belén Martínez Prieto
Universidad de Oviedo

De Madrid a Bolonia pasando por Pittsburg	55
Raúl Murciano, Pedro J. Lara, David Aauri <i>Universidad Europea de Madrid</i>	
Tema estratégico: Formación para la profesión.....	63
Reflexiones sobre la enseñanza de la Auditoría de Sistemas de Información en las escuelas de informática	65
Eduardo Fernández-Medina, Mario Piattini <i>Universidad de Castilla - La Mancha</i>	
Sobre la formación no técnica en la Ingeniería Informática.....	73
Josep M. Basart Muñoz <i>Universitat Autònoma de Barcelona</i>	
La Función TSI en las organizaciones: una evaluación formativa para detección de "gaps" críticos.....	79
Ferrán Virgós Bel <i>Universitat Politècnica de Catalunya</i>	
Un modelo de diseño curricular de Informática orientado a la obtención de competencias	89
Félix Buendía, Juan Carlos Cano, Julio Sahuquillo, Juan Luis Posadas, Juan Miguel Martínez, José Vicente Benlloch <i>Universidad Politécnica de Valencia</i>	
Aprendizaje cooperativo: implantación de esta técnica en dos asignaturas reformadas y evaluación de resultados.....	95
Antoni Pérez-Poch <i>Universitat Politècnica de Catalunya</i>	
Preparamos a nuestros alumnos para el mercado laboral: Integración de la terminología profesional en las aulas	103
María José García García, Luis Fernández Sanz, Maite Villalba de Benito <i>Universidad Europea de Madrid</i>	

Arquitectura de ordenadores	109
Java, herramienta docente en asignaturas del área de Arquitectura de Computadores.....	111
J. A. Álvarez Bermejo, J. F. Sanjuán Estrada, D. P. Morales Pastor <i>Universidad de Almería</i>	
Práctica de optimización para asignaturas de Estructura de los Computadores	119
M. Anguita, F. J. Fernández, A. F. Díaz, A. Cañas, A. Prieto <i>Universidad de Granada</i>	
Automatización del problema de asignación de estados en el diseño de sistemas secuenciales síncronos.....	127
Juan Fco. Sanjuán Estrada, I. García Fernández, J. A. Álvarez Bermejo <i>Universidad de Almería</i>	
El proyecto de investigación: Un complemento eficaz en la docencia de Arquitectura de Computadores.....	135
José M. García, Manuel E. Acacio <i>Universidad de Murcia</i>	
Internet y herramientas software de apoyo a la docencia en Estructura y Tecnología de Computadores I.....	143
José Luis Poza, Juan Luis Posadas, Juan Carlos Cano, Salvador Petit <i>Universidad Politécnica de Valencia</i>	
La Aritmética del Computador en el Diseño Curricular	151
Julio Sahuquillo, José Flich, Salvador Petit, Juan Carlos Cano <i>Universidad Politécnica de Valencia</i>	
Bases de Datos.....	159
Herramienta docente para consultas en Oracle	161
María Belén Vaquerizo García, Antonio Eduardo Renedo Mena <i>Universidad de Burgos</i>	

Calidad y evaluación de la docencia	169
Decálogo para el profesor novel.....	171
Faraón Llorens, Rosana Satorre <i>Universidad de Alicante</i>	
La colaboración en el Jenui, a quién nos parecemos y a quién no	179
José Miró, Ricardo Alberich <i>Universitat de les Illes Balears</i>	
Objetivos formativos y estrategias docentes para el primer curso de las ingenierías informáticas	187
Fermín Sánchez, Ricard Gavalda <i>Universitat Politècnica de Catalunya</i>	
Evaluación del alumnado.....	195
Retroalimentación a los alumnos tras los exámenes finales: un caso práctico	197
Agustín Cernuda del Río <i>Universidad de Oviedo</i>	
La copia de prácticas de programación: el problema y su detección	203
Pedro J. Clemente, Alberto Gómez, Julia González <i>Universidad de Extremadura</i>	
Sistema tutor avanzado.....	211
David Vernet, Xavier Canaleta <i>Universitat Ramon Llull</i>	

Fundamentos Teóricos de la Informática	217
ToTalf: Un Tutorial On-line para Teoría de Automatas y Lenguajes Formales.....	219
Román Sosa, Francisco de Sande <i>Universidad de la Laguna</i>	
¿Hay lógica en la situación actual de las titulaciones informáticas?.....	227
José Emilio Labra Gayo <i>Universidad de Oviedo</i>	
Ingeniería del software.....	235
La Experimentación en la docencia de Ingeniería del Software	237
Félix García, Manuel Serrano, José A. Cruz-Lemus, Marcela Genero, Coral Calero, Mario Piattini <i>Universidad de Castilla – La Mancha</i>	
Estudio de la Distribución Docente de Pruebas del Software y Refactoring para la Incorporación de Metodologías Ágiles	247
Raúl Marticorena, Carlos López, Yania Crespo <i>Universidad de Burgos</i>	
BDeIS: Integración, mejora e investigación en el aprendizaje de las Bases de Datos y la Ingeniería del Software	255
Manuel Palomar, Paloma Moreda, Andrés Montoyo, Rafael Muñoz, Patricio Martínez, Jaime Gómez, Eva Gómez, Armando Suárez, Cristina Cachero, J. Carlos Trujillo <i>Universidad de Alicante</i>	
Del Diseño a la Implementación del Software: Una metodología de cohesión.....	263
Cristina Cachero, Otoniel López, María José Durá <i>Universidad de Alicante</i>	

Aportación del paradigma orientado a agentes en el desarrollo de tutores de e-learning.....	271
Andrés Castillo, Oscar Sanjuán, Luis Joyanes <i>Universidad Pontificia de Salamanca</i>	
Métodos pedagógicos innovadores.....	279
Experiencia inter-universitaria de prácticas en Sistemas Abiertos	281
Pedro García López, Jesús Egea Payá, Eduardo Martínez Graciá <i>Universidad de Murcia</i>	
Experiencias en innovación docente: aspectos positivos y negativos de un caso real.....	289
A. Martí Campoy, J. C. Campelo Rivadulla, R. Ors Carot <i>Universidad Politécnica de Valencia</i>	
¿Por qué no enseñamos a aprender cooperativamente?	297
V. Javier Traver, Joan A. Traver <i>Universidad Jaume I</i>	
Aprendizaje Basado en Problemas: Una experiencia interdisciplinar en Ingeniería Técnica en Informática de Gestión	305
Raquel Lacuesta, Carlos Catalán <i>Universidad de Zaragoza</i>	
Herramientas de soporte a la evaluación docente basadas en tecnologías web.....	313
Miguel Sánchez López, Alberto González Téllez <i>Universidad Politécnica de Valencia</i>	
Foros Virtuales Colaborativos en Línea Aplicados a Procesos de Tutorización.....	321
Rafael Romero, Maximiliano Sáiz, José L. Verdú, José L. Vicedo <i>Universidad de Alicante</i>	

Tratando de fomentar la motivación del estudiantado	329
Mercedes Marqués, Vicente Ramón Tomás, Ismael Sanz <i>Universidad Jaume I</i>	
Difusión e intercambio de material docente: soporte en las sedes web	337
Iker Azpeitia, Oscar Díaz, Jesús Ibáñez <i>Universidad del País Vasco</i>	
Laboratorio no presencial. Un ejemplo para la asignatura de Base de Datos	345
Miguel Ángel Quintana Suárez <i>Universidad de las Palmas de Gran Canaria</i>	
Multimedia e Informática Gráfica.....	353
Propuesta docente para gráficos por ordenador en el nuevo contexto europeo de educación.....	355
Rafael Molina Carmona, Juan Antonio Puchol García <i>Universidad de Alicante</i>	
Programación de mundos virtuales: experiencia docente en un curso no reglado	363
Fidel Aznar, Miguel Ángel Lozano, Rafael Molina, Pablo Suau, Diego Viejo <i>Universidad de Alicante</i>	
Programación, algoritmos y estructuras de datos.....	369
Aproximación paralela a la técnica Divide y Vencerás	371
A. Dorta, L. García, J. R. González, C. León, C. Rodríguez <i>Universidad de la Laguna</i>	
Generación de recursos educativos para la enseñanza de la asignatura de "Estructuras de datos y de la información" en un campus virtual.....	379

Antonio Sarasa <i>Universidad Complutense de Madrid</i>	
Un modo de entender la programación	387
Rosana Satorre, Patricia Compañ, Faraón Llorens <i>Universidad de Alicante</i>	
Una propuesta para una primera asignatura de Algoritmia	393
Francisco Palomo Lozano, Inmaculada Medina Buló <i>Universidad de Cádiz</i>	
Un enfoque semiformal para la introducción a la programación	401
Jesús García Molina <i>Universidad de Murcia</i>	
EDApplets: Estructuras de datos y Técnicas Algorítmicas en la Web.....	409
F. Almeida, V. Blanco, L. M. Moreno <i>Universidad de la Laguna</i>	
Propuesta de créditos ECTS para la asignatura de Programación de primer curso de Ingeniería	417
Xavier Canaleta, David Vernet <i>Universitat Ramon Llull</i>	
Aplicación de JML a las prácticas de programación con Java.....	425
Antonio David Gómez Morillo, Antonio Menchén Peñuela <i>Universidad de Sevilla</i>	
Robótica e informática industrial	433
Entornos para prácticas de control y comunicaciones en asignaturas de informática industrial y domótica.....	435
Andrés Fuster Guilló, Francisco J. Ferrández Pastor, Jorge Azorín López <i>Universidad de Alicante</i>	

Ambientalización Curricular de los Estudios de Informática I
experiencia de la UPC

Yolanda Bolea, Antoni Grau, Joan Domingo, Juan Gá
Martínez
Universitat Politècnica de Catalunya

Sistemas operativos

Docencia de Semáforos y Monitores con BACI: un ejemplo d
guiado

V. Galiano, O. López, M. Martínez, H. Migallón, P. P
Universidad Miguel Hernández

Propuesta de contenido de prácticas en la materia de Sistemas
Miguel Riësko Albizu, Marián Díaz Fondón
Universidad de Oviedo

Tecnologías de la información en la gestión empresarial ...

Un método para el diseño de la programación docente de una
utilizando distintos modelos curriculares de referencia

Edmundo Tovar, José Carrillo
Universidad Politécnica de Madrid

Telemática

Diseñando servicios móviles en el laboratorio de redes.....

Vicente Galiano, Otoniel López, Miguel Martínez, He
Pablo Piñol, David Ubeda
Universidad Miguel Hernández

Recursos Docentes

- Web dinámica para el aprendizaje del cálculo relacional 493
 Diego Carpintero Millán, César Ignacio García Osorio, Jesús M. Maudes
 Raedo
Universidad de Burgos
- CCEDAF: Codificador Combinatorio de Estados para el Diseño de
 Automatas Finitos..... 497
 Juan Fco. Sanjuán Estrada, I. García Fernández, J. A. Álvarez Bermejo
Universidad de Almería
- Simulador de una Unidad de Control microprograma 501
 Aitor Jáuregui Vidal, Txelo Ruiz Vázquez, Izaskun Etxeberria Uztarroz
Universidad del País Vasco
- SIMDE: Un Simulador para el Apoyo Docente en la Enseñanza de las
 Arquitecturas ILP con Planificación Dinámica y Estática 505
 I. Castilla, L. Moreno, J. Sigut, C. González, E. J. González
Universidad de la Laguna
- Simulador de un robot autónomo para las prácticas de ensamblador de la
 asignatura de Estructura de Computadores 509
 Pedro A. Castillo, A. Cañas, A. Prieto, Juan J. Castillo
Universidad de Granada
- Aprendiendo mediante juegos: Experiencia de una competición de
 juegos inteligentes 513
 O. Colomina, P. Compañ, R. Satorre, F. Aznar, P. Suau, R. Rizo
Universidad de Alicante



conferencias

La Experimentación en la docencia de Ingeniería c

Félix García, Manuel Serrano, José A. Cruz-Lemus, Marcela Genero

Mario Piattini

Grupo Alarcos

Escuela Superior de Informática

Universidad de Castilla-La Mancha

13001 Ciudad Real

e-mail: {Felix.Garcia, Manuel.Serrano, JoseAntonio.Cruz, Marcela.Genero, Coral.Calero, Ma

Resumen

Los estudios empíricos en Ingeniería del Software son fundamentales para la validación de diversos métodos, técnicas, herramientas, etc., y los alumnos juegan un papel fundamental a la hora de llevarlos a cabo. Estos estudios no permiten obtener beneficios centrados exclusivamente en los aspectos de investigación, sino que es muy importante considerar también sus beneficios en la docencia. En este artículo se estudia la aplicación de experimentos controlados en cursos de Ingeniería de Software, destacando los beneficios que estos estudios aportan a los alumnos y a los docentes e investigadores que los llevan a cabo. Además, se presentan los resultados obtenidos en la realización de tres experimentos en cursos de ingeniería del software. Como consecuencia de llevar a cabo estos experimentos se han obtenido importantes beneficios pedagógicos.

1. Introducción

Uno de los problemas de la Ingeniería del Software consiste en que a menudo se proponen una gran diversidad de métodos, lenguajes, herramientas, entornos, etc., de los que no se demuestra su utilidad práctica. Por ello, los métodos empíricos han centrado la atención de la comunidad científica en la ingeniería del software durante los últimos años. Mediante los métodos empíricos es posible evaluar nuevas aportaciones en la ingeniería del software antes de que sean introducidas en los procesos software de las empresas [10].

Los estudios empíricos más comúnmente utilizados en la Ingeniería del Software son: experimentos controlados, casos de estudio y encuestas, los cuales difieren fundamentalmente

en sus objetivos y restricción académica. Los estudios académicos son significativos tanto desde el punto de vista del investigador como del docente [1].

A la hora de llevar a cabo los estudios, los alumnos juegan un papel muy importante antes de realizar estos experimentos en entornos industriales (lo que requiere de tiempo, esfuerzo y recursos). En algunas ocasiones los investigadores realizan estudios piloto con alumnos académicos [4]. De hecho, los estudios académicos constituyen la base para los estudios profesionales [12]. Por ello, los estudios en entornos académicos tienen una importancia y permiten obtener resultados significativos de cara a su aplicación en entornos industriales. Bajo estas condiciones las diferencias entre los estudios académicos y profesionales son pequeñas; en ciertos experimentos no realizados en entornos industriales, por ello se justifica la experimentación con alumnos [2].

Además, los estudios académicos constituyen una aportación importante a la investigación pedagógica en cursos de Ingeniería del Software, por lo que se establece una relación entre la investigación y la docencia fundamental para analizar [4].

Hoy en día los alumnos de Ingeniería del Software emplean métodos empíricos (sobre todo los desarrollados en entornos norteamericanos y británicos) para validar sus hipótesis. Por ello una gran cantidad de métodos empíricos han podido ser validados empíricamente. Esto se refleja en numerosas publicaciones, pero, sin embargo, muchas de estas

centran en presentar los beneficios que los estudios han aportado a la investigación, dejando de lado a los alumnos, lo que incluso hace pensar en ocasiones, que los alumnos han sido "utilizados" de forma egoísta para obtener resultados en la investigación. Por ello es fundamental abordar los beneficios que la experimentación aporta desde el punto de vista pedagógico y aportar estos beneficios a los alumnos cuando se planifican y se llevan a cabo experimentos en entornos académicos.

El principal objetivo de este trabajo es el de estudiar la aplicación de experimentos en cursos de Ingeniería de Software, destacando los beneficios que estos estudios aportan, tanto desde el punto de vista pedagógico como investigador.

A continuación, en el apartado 2 se identifican los distintos beneficiarios de los experimentos y las ventajas que estos estudios les aportan, sobre todo para el investigador, el docente y los alumnos. En el apartado 3 se describen las fases que hay que considerar a la hora de llevar a cabo experimentos. Los resultados de diversos experimentos llevados a cabo con alumnos en distintos cursos de Ingeniería de Software, se presentan en el apartado 4. Finalmente se presentan las conclusiones obtenidas y las consideraciones para el futuro.

2. Beneficiarios de la Experimentación en Ingeniería del software.

Cuando se desarrollan experimentos se pueden identificar cuatro beneficiarios principales de los resultados. Cada uno de los cuales tendrán diferentes puntos de vista [4]:

- **Investigador.** Es el encargado de planificar y llevar a cabo el experimento. Su objetivo es demostrar la utilidad práctica de su propuesta u obtener conclusiones preliminares para llevar a cabo experimentos en entornos industriales.
- **Docente.** Es el profesor responsable de la asignatura o grupo de alumnos, que constituyen el contexto en el que se lleva a cabo el experimento. Su principal objetivo es enseñar los conocimientos y habilidades relacionadas con los estudios empíricos realizados y que van a ser de utilidad a los alumnos en su trabajo como futuros profesionales.

- **Alumnos.** Son los sujetos utilizados en el experimento. Su objetivo es aprender técnicas y habilidades que les puedan servir como futuros profesionales.
- **Empresa.** Las empresas de software son las beneficiarias últimas de los estudios empíricos. Como resultado de dichos estudios, las empresas pueden adoptar nuevos métodos o tecnologías que influirán en la mejora de sus procesos software y en definitiva les permitirán obtener beneficios económicos a medio o largo plazo.

En el contexto académico es de especial relevancia establecer las principales ventajas de los estudios empíricos desde el punto de vista del docente e investigador, que en muchas ocasiones son la misma persona, y sobre todo el beneficio que se aporta a los alumnos. En la Tabla 1 se resumen las ventajas más significativas de llevar a cabo estudios empíricos con alumnos de acuerdo al análisis realizado en [4]:

Beneficiario	Beneficios
Docente	<ul style="list-style-type: none"> • Nueva forma de formar a los alumnos respecto de la enseñanza tradicional, alumno • Fomento de la participación en grupo de los alumnos en determinados estudios empíricos • Mejora de la comunicación con los alumnos • Nuevas formas de evaluación de los alumnos en situaciones en las que no tienen el estrés típico de un examen formal • Introducir la ingeniería del software empírica como parte de la enseñanza en la ingeniería del software
Investigador	<ul style="list-style-type: none"> • Evidencia preliminar para aceptar o rechazar hipótesis • Demostrar a las empresas software la relevancia de la investigación y la utilidad de llevar a cabo estudios empíricos en las propias empresas • Prever los recursos necesarios para realizar experimentos en entornos industriales y preparar el material necesario del experimento para realizarlo en la industria • Formación de investigadores noveles en el desarrollo de estudios

Beneficiario	Beneficios
	empíricos
Alumno	<ul style="list-style-type: none"> • Formación en materias complementarias a la formación de grado • Conciencia de nuevos problemas a resolver en el software en general y en las industrias en particular • Mejor autoevaluación de cuál es su nivel en determinados temas de ingeniería del software que en las clases tradicionales más centradas en los aspectos teóricos • Percepción de las ventajas de usar métodos empíricos en la ingeniería del software • Preparación para su futuro profesional en el que en muchas ocasiones serán sometidos a cuestionarios, informes, encuestas, etc.

Tabla 1. Beneficios de los estudios empíricos en entornos académicos desde la perspectiva del docente, del investigador y de los alumnos

En la práctica, es muy importante conocer los efectos negativos de llevar a cabo experimentos en entornos académicos, con el fin de que la planificación de los experimentos sea suficientemente cuidadosa para que estos efectos no se produzcan [4]:

- Desde el punto de vista del **investigador** los inconvenientes de realizar estudios empíricos con alumnos son el esfuerzo necesario para prepararlos, ya que nunca deben olvidarse de los aspectos pedagógicos al preparar el estudio. Otro problema son las amenazas a la validez del estudio, tema que se aborda brevemente en el apartado 3.
- Desde el punto de vista del **docente** los problemas que pueden surgir son la necesidad de motivar al docente para llevar a cabo el experimento, ya que le supone un esfuerzo de formación mucho mayor para esa clase que si se tratara de una clase normal y además debe motivar a los propios alumnos creando el ambiente necesario en la clase. El docente debe tener capacidad de atender cualquier duda de los alumnos. Estos problemas se solucionan en gran medida si el docente y el investigador son la misma persona, situación

que es deseable a la hora de realizar estudios empíricos con alumnos.

- Desde el punto de vista de los **alumnos** pueden surgir problemas derivados de pérdida de tiempo, si se trata de experimentos que requieren una formación extensa de los alumnos que les hace perder varias clases en lugar de aprovecharlas formándoles en temas más interesantes o útiles para su futuro profesional. Otro problema potencial es que el resultado del experimento demuestre que la técnica o método que acaban de aprender no es efectivo, aunque en este caso también se puede encontrar la parte pedagógica, consistente en demostrarles que una tecnología o técnica no se puede aceptar sin evaluarla en la práctica por muy nueva que sea. Además, en este último caso se pueden intentar hallar las causas del resultado.

En definitiva, es fundamental considerar las ventajas e inconvenientes comentados en la planificación de estudios empíricos, con el fin de que las ventajas sean percibidas por los alumnos y se puedan evitar en la medida de lo posible los problemas que de ello se puedan derivar, como la desmotivación o el descontento. Los autores de este artículo han llevado a cabo como docentes/investigadores experimentos con alumnos en diferentes cursos de la Escuela Superior de Informática de Ciudad Real. Los resultados de dichos experimentos centrados en los aspectos más pedagógicos se describen en el apartado 4. A continuación se describen de forma resumida las principales características y aspectos a considerar a la hora de llevar a cabo experimentos controlados con alumnos.

3. Proceso Experimental

En este apartado se describen los factores esenciales que hay que considerar a la hora de llevar a cabo una buena planificación y realización de experimentos controlados, con el fin de obtener resultados que sean creíbles [11],[17],[13], [3], [12].

Las ventajas de los experimentos es que pueden determinar las situaciones en las que ciertas afirmaciones son verdaderas y pueden proporcionar el contexto en el que ciertos estándares, métodos y herramientas son

recomendables. Si el experimento se realiza adecuadamente, seremos capaces de sacar conclusiones acerca de las relaciones entre la causa y el efecto para la cual formulamos la hipótesis (la cual queremos corroborar mediante la realización de experimentos). Los experimentos necesitan ser planeados cuidadosamente si queremos que nos proporcionen resultados útiles y significativos. Por ello es necesario seguir un proceso experimental como el que se propone en [17] que consta de las siguientes etapas:

1. **Definición.** Este es el primer paso, en el definiremos el experimento en términos del problema y los objetivos.
2. **Planificación.** donde determinaremos el diseño del experimento y consideraremos la instrumentación del mismo.
3. **Operación.** Esta es la fase en la que se lleva a cabo el experimento y se recogen los datos empíricos.
4. **Análisis e interpretación.** Los datos recogidos son analizados utilizando técnicas estadísticas. Además se interpretan los resultados obtenidos.
5. **Evaluación de la validez.** En este paso evaluaremos los aspectos que pueden amenazar a la validez del experimento. Para ello se consideran se estudian cuatro tipos de validez:
 - **Validez de constructo.** Define hasta qué punto las medidas utilizadas para medir las variables independientes y dependientes, miden fielmente los conceptos que intentan medir.
 - **Validez interna.** Se refiere al grado de confiabilidad con el que podemos asegurar la relación causa-efecto entre las variables independientes y dependientes.
 - **Validez externa.** Consiste en el poder de generalización de los resultados, no sólo para el entorno donde se ha desarrollado el experimento sino también en otros entornos.
 - **Validez de conclusión.** Define hasta qué punto las conclusiones obtenidas son estadísticamente válidas.
6. **Presentación y difusión.** En esta última fase se elabora un informe sobre los resultados para que los mismos sean difundidos con el objetivo de que otros investigadores puedan replicar el experimento.

4. Experimentación con alumnos e cursos de Ingeniería de Software.

En este apartado se presentamos resultados de tres experimentos controlados llevados a cabo en asignaturas de Ingeniería de Software. Los experimentos tuvieron lugar en horario normal de clase, su realización era voluntaria y motivó especialmente a los estudiantes para su realización, destacando los beneficios de los experimentos les proporcionó como futuros profesionales. Para conseguir evitar las posibles amenazas a la validez de los experimentos, tomamos una serie de medidas:

- Los sujetos que se convocan para hacer cada experimento tenían una experiencia y unos conocimientos parecidos.
- Los dominios de los diagramas eran lo suficientemente sencillos y comunes para que no existiese ningún problema para entenderlos.
- Para evitar los efectos de aprendizaje los esquemas fueron entregados a cada sujeto en un orden diferente.
- Los sujetos que realizaron el experimento era la primera vez que realizaban un experimento de este tipo, por lo que los efectos de la persistencia están atenuados.
- Los sujetos estaban motivados por los ejercicios formaban parte de los conocimientos que debía adquirir en su formación.
- No se permitió que los sujetos hablaran entre ellos durante la prueba ni se pudieran copiar los resultados unos de otros.
- Todas las dudas fueron resueltas por la persona que supervisaba el experimento.
- Los sujetos no tenían conocimiento, a priori, de los aspectos que se pretendían estudiar ni cuáles eran las hipótesis que se estaban planteando.

4.1. Experimento sobre Métodos de Diseño de Almacenes de Datos

Definición

Se han propuesto diversos métodos de diseño de almacenes de datos basados en los diagramas de estrella, ya que estos diagramas aumentan la

eficacia en la comprensión de los esquemas de los almacenes de datos. Aunque esta afirmación es ampliamente aceptada, no se ha demostrado empíricamente que sea cierta, por lo que consideramos adecuado realizar un experimento para poder comprobarlo.

El objetivo de este experimento [14] era determinar si los diseños de almacenes de datos basados en diagramas de estrella son más comprensibles que aquellos realizados basándose en el diseño tradicional de bases de datos.

Planificación

El experimento se realizó con 11 alumnos de la Escuela Superior de Informática de Ciudad Real (UCLM). Todos nuestros sujetos tenían conocimientos de diseño y uso de bases de datos, además poseían conocimientos del diseño y uso de los almacenes de datos pues habían recibido esta información como parte de sus estudios de doctorado.

El experimento constaba de tres diagramas diseñados según el diseño tradicional de bases de datos y otros tres diagramas semánticamente equivalentes a los anteriores usando diagramas de estrella, sobre los que había que realizar unas consultas utilizando SQL. Los diagramas representaban problemas reales y sus dominios eran lo suficientemente sencillos como para ser comprendidos fácilmente.

El objetivo del experimento era analizar si había diferencia en los tiempos de respuesta entre los diagramas semánticamente equivalentes para analizar: alguno resultaba más complejo.

Operación

Antes de realizar el experimento se explicó el tipo de problemas que debían resolver, cómo se debía contestar a las preguntas y que material se estaba proporcionando para su realización.

Análisis e Interpretación

Según los objetivos del estudio y la configuración del experimento usamos una prueba ANOVA de medidas repetidas univariante y como resultado de este análisis estadístico se llegó a la conclusión de que no hay diferencia en la comprensión de los esquemas debido al método de diseño utilizado.

También se dedujo que el uso de esquemas tampoco influye en los mismos.

Conclusiones

Como conclusión del experimento se dedujo que no hay diferencia de los esquemas debido al uso de diagramas de estrella en el almacén de datos utilizando diagramas de estrella o los diagramas tradicionales.

Cuando comunicamos los resultados a los sujetos, intentamos sacar algunos resultados que obteníamos por el uso de diagramas de estrella es ampliamente utilizado por los alumnos. Dado que los resultados de los experimentos no fueron ampliamente aceptados y no de los resultados de los experimentos. No obstante, intentamos ver cuáles de los resultados que podían haber afectado a los resultados de los experimentos obtuvimos las siguientes:

- El tamaño de los esquemas no era muy grande.
- Los dominios eran los mismos que los conocidos como para no tener problemas de complejidad y la comprensión de los mismos.
- Los sujetos no estaban tan motivados para trabajar con diagramas de estrella como para poder afectar al tiempo de las consultas.

Los beneficios obtenidos de este experimento fueron variados. Los alumnos se veían dentro de un entorno empírico y sepan cómo hacer los experimentos. Además los alumnos dudaban de los prejuicios y a las conclusiones que los resultados obtenidos esperados.

4.2. Experimento de Modelos de Software

Definición

El modelado y la medición de software se han convertido en aspectos fundamentales para promover el desarrollo de los mismos. Con el fin de medir la complejidad estructural de

procesos software se han definido un conjunto representativo de métricas [6]. Para validar las métricas propuestas de llevó a cabo un experimento cuyo objetivo fue establecer la relación entre la complejidad estructural de los modelos de procesos y su mantenibilidad a través de dos de sus subcaracterísticas: la entendibilidad y la modificabilidad. La mantenibilidad de un modelo de procesos es la facilidad con la que puede ser mantenido. Así, modelos de procesos software difícilmente mantenibles, pueden afectar a la ejecución de los proyectos (duración, coste, etc.) y a la calidad final de los productos obtenidos.

Planificación

El experimento se realizó con 45 alumnos de ingeniería técnica en informática de gestión y 41 de ingeniería técnica en informática de sistemas de tercer curso de la Escuela Superior de Informática de Ciudad Real. Todos los sujetos tenían conocimientos de modelado del producto pero no del modelado del proceso software. Por ello se les impartió una clase en la que se les explicó la importancia de la gestión efectiva de los procesos software para mejorar la calidad de los productos finales, haciendo especial hincapié en la influencia que en ello tiene el modelado del proceso. Como lenguaje de modelado del proceso software se explicó el lenguaje SPEM [16].

El material del experimento constaba de diez modelos de procesos software desarrollados a partir de diversos estándares y metodologías existentes en la bibliografía. El objetivo del experimento consistía en evaluar si la complejidad de los modelos de procesos (variable independiente) influye en la entendibilidad y en la modificabilidad de los modelos (variables dependientes).

Los objetos usados fueron diez modelos de procesos. Cada modelo incluía una hoja de ejercicios compuesta por una sección que podía consistir en cinco cuestiones relacionadas con los modelos de procesos o en cuatro ejercicios consistentes en realizar modificaciones sobre los modelos. En total cada sujeto tenía cinco modelos con cuestiones relacionadas con la entendibilidad y cinco modelos con cuestiones relacionadas con la modificabilidad.

La variable independiente fue medida mediante las métricas propuestas, mientras que las

variables dependientes fueron medidas calculando el tiempo que los sujetos tardaron en responder las cuestiones relacionadas con la entendibilidad de cada modelo (tiempo de entendibilidad) y mediante el tiempo que los sujetos emplearon en realizar las modificaciones que se requerían para cada modelo (tiempo de modificabilidad) respectivamente. Nuestra suposición es que cuánto menos tiempo se emplea en entender y modificar un modelo de procesos, más fácil será mantenerlo.

Operación

En primer lugar se impartió una sesión preparativa previa en la que se resolvió un ejemplo similar al material que luego se les entregaría.

Análisis e Interpretación

Utilizando el coeficiente de correlación de Spearman se analizó si existía correlación entre las métricas para la complejidad estructural de los modelos de procesos software y los tiempos de entendibilidad y de modificabilidad de los mismos. Como resultado se obtuvo que varias de las métricas estaban correlacionadas con dichos tiempos, por lo que consideramos que —en cierta manera— son válidas.

Conclusiones

Desde el punto de vista del investigador se llegó a la conclusión de que las métricas definidas eran —en cierta manera— válidas, y podrían ser utilizadas como indicadores del tiempo de entendimiento y mantenimiento de los modelos de procesos software, además de servir para poder comparar modelos semánticamente equivalentes.

Desde el punto de vista del alumno, la realización del experimento en la asignatura de Ingeniería del Software de tercer curso, permitió impartir en dicha asignatura un tema relacionado con el proceso software. En dicho tema se estudió la definición del Proceso Software, sus elementos más importantes, la importancia de la gestión de su calidad y se profundizó en el modelado de los mismos. Ello permitió a los alumnos tener una visión más amplia de la asignatura de Ingeniería del Software, ya que el proceso software sólo se estudia al tratar los modelos de ciclo de vida del software, y durante el curso aplican técnicas del

modelado del producto, pero no del proceso. Por ello consideramos que el experimento les beneficia como futuros profesionales.

Además, también hay que considerar los beneficios pedagógicos que podemos obtener cuando se les comunican los resultados. Para ello, se imparte una clase en la que por un lado se les hace ver la importancia de realizar estudios empíricos y por otro, se les motiva comparando sus resultados con los resultados del mismo experimento llevado a cabo en un entorno industrial, que han sido similares [7]. Ello demuestra que en determinadas circunstancias no existen diferencias significativas entre los alumnos de últimos cursos de carrera y los profesionales.

4.3. Experimento de Diagramas de Clases UML

Definición

Con la idea de que cuánto más complejos sean los diagramas de clases UML más difícil será entenderlos y luego mantenerlos se definieron un conjunto de métricas para medir la complejidad estructural y tamaño de tales diagramas [8]. Para validar empíricamente dichas métricas se llevó a cabo un experimento controlado. Por ello, el principal objetivo de este experimento era averiguar si realmente existe correlación entre la complejidad estructural y el tamaño de los diagramas de clases UML y la entendibilidad y mantenibilidad de los mismos.

Planificación

Los sujetos que realizaron el experimento fueron 24 alumnos del tercer curso de la Ingeniería Informática, en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha, en Ciudad Real. Dichos alumnos estaban cursando el primer curso de Ingeniería de Software de la carrera. Por lo que no los podemos considerar expertos, aunque también habían cursado otras asignaturas de programación orientada a objetos en cursos previos.

Las variables independientes eran la complejidad estructural y el tamaño de los diagramas de clases UML, medidas a través de un conjunto de métricas definidas en [8].

Las variables dependientes eran la entendibilidad y la mantenibilidad de los diagramas de clases UML, medidas a través del tiempo que los sujetos emplearon para realizar las tareas requeridas en el experimento.

El experimento consistió en 9 diagramas de clases UML sobre distinto dominio de aplicación, pero lo suficientemente general para ser fácilmente comprendido por los sujetos. Las tareas a realizar fueron de dos tipos:

- Contestar el cuestionario adjunto a cada diagrama de clases, que consistía en cinco preguntas, y anotar el tiempo que tardaban en contestarlas, al que llamamos tiempo de entendimiento (expresado en segundos).
- Modificar cada diagrama de clases de acuerdo a cuatro nuevos requerimientos y anotar el tiempo utilizado, al que llamamos tiempo de mantenimiento (expresado en segundos).

Se eligió un diseño experimental intrasujetos, lo que significa que todos los sujetos debían realizar todas las mismas tareas sobre los 9 diagramas de clases UML.

Operación

Antes de realizar el experimento el profesor que supervisaba el experimento realizó una sesión de entrenamiento, en la que se les entregó a los sujetos un diagrama de clases con una serie de preguntas y tareas de mantenimiento, similares a las que luego deberían realizar en el experimento.

Se les comentó a los alumnos que esto les serviría para adquirir práctica en el diseño de diagramas de clases de UML, ya que no sólo es importante saber diseñarlos completamente, sino también saber entenderlos y poder modificarlos.

Como resultado obtuvimos los datos del cálculo de las métricas de la complejidad estructural y el tamaño de cada diagrama y los tiempos de entendibilidad y modificación que cada sujeto ha necesitado para realizar las tareas requeridas sobre cada diagrama. Adicionalmente se han comprobado las tareas realizadas para establecer unos índices de corrección y completitud.

Análisis e Interpretación

Analizando los datos recogidos mediante el coeficiente de correlación de Spearman obtuvimos que la mayoría de las métricas definidas estaban correlacionadas con el tiempo de entendimiento y mantenimiento de los diagramas de clases UML.

Conclusiones

Desde el punto de vista del investigador se llegó a la conclusión de que las métricas definidas eran – en cierta manera – válidas, y podrían ser utilizadas como indicadores del tiempo de entendimiento y mantenimiento de los diagramas de clases UML. Además de servir para poder comparar diagramas semánticamente equivalentes.

Desde el punto de vista del alumno la realización del experimento les permitió obtener cierta experiencia en el entendimiento y modificación de diagramas de clases UML, tareas que podrían tener que realizar tanto en los exámenes como en el desempeño de la profesión.

En una clase posterior, se les explicó a los alumnos el objetivo de investigación que perseguía el experimento. Se les explicaron brevemente las métricas definidas, las hipótesis planteadas y los resultados obtenidos. Por lo que creemos, que además pudieron aprender un tema que aún no habían visto en Ingeniería de Software que fue la utilización de métricas.

5. Conclusiones

En este artículo se ha abordado la realización de experimentos en entornos académicos, destacando las ventajas que estos estudios aportan tanto desde el punto de vista del docente/investigador como del punto de vista del alumno. Además, se han presentado los resultados obtenidos en la realización de tres experimentos en cursos de ingeniería del software. Estos resultados confirman los beneficios docentes obtenidos, destacando los siguientes:

- Formación en los principios básicos de la experimentación software.
- Aprendizaje de los procesos software, su modelado y tecnología.
- Importancia del uso de métricas para evaluar cuantitativamente técnicas, métodos y estándares en Ingeniería del Software.
- Evaluación de distintas técnicas de diseño.

- Práctica en el diseño de diagramas de clases con UML.
- Realimentación a partir de los resultados de los experimentos.

El entusiasmo mostrado por la mayoría de los alumnos al realizar el experimento y su interés por conocer los resultados obtenidos nos lleva a pensar que este tipo de experimentos son muy beneficiosos y se deberían realizar siempre que sea posible en cursos de Ingeniería del Software, tal y como ya se está realizando en otros centros de estudio internacionales [5], [10].

No obstante a la hora de llevar a cabo experimentos en cursos de ingeniería del software siempre hay que considerar los beneficios que se pueden aportar a los alumnos y además hay que tener en cuenta diversas consideraciones éticas [15], para que como resultado del experimento no se produzca desmotivación y descontento por parte de los alumnos.

Referencias

- [1] Baresi, L., Morasca, S. and Paolini, P. Estimating the Design Effort of web Applications. *Proceedings of the 9th International Software Metrics Symposium (METRICS'00)*, pp. 62-71, 2003.
- [2] Basili V., Stahl F. and Lanubile F. Building Knowledge through Families of Experiments. *IEEE Transactions on Software Engineering*, Vol. 25 N° 4, pp. 435-437, 1999.
- [3] Briand, L., Arisholm, S., Counsell, F., Houdek, F. and Thévenod-Fosse, P. (2000). Empirical studies of Object-Oriented Artefacts, Methods, and Processes: State of the Art and Future Directions. *Empirical Software Engineering*, Vol. 4 N° 4, pp. 387-404, 2000.
- [4] Carver, J., Jasheri, L., Morasca, S. y Shull, F. Issues in Using Students in Empirical Studies in Software Engineering Education. *Proceedings of the 9th International Software Metrics Symposium (METRICS'03)*, pp. 239-251, 2003.
- [5] Carver, J., Jasheri, L., Morasca, S. y Shull, F. Using Empirical Studies during Software Courses. *Experimental Software Engineering Research Network 2001-2003*. LNCS 2765, pp. 81-103, 2003.
- [6] García, F., Ruiz, F. y Piattini, M. "Medición del Proceso Software", *VIII Jornadas de Ingeniería del Software y Bases de Datos*, Actas de las Jornadas E. Pimentel, N. Brisaboa, J. Gómez (eds), Alicante (España), pp. 303-314, 2003.
- [7] García, F., Ruiz, F. y Piattini, M. Definition and Empirical Validation of Metrics for Software Process Models. *5th International Conference on Product Focused Software Process Improvement (PROFES'2004)*. Lecture Notes in Computer Science (LNCS 3009), pp. 146-158, 2004.
- [8] Genero, M. Defining and Validating Metrics for Conceptual Models. Ph.D. Thesis Department of Computer Science, University of Castilla-La Mancha, 2002.
- [9] Höst, M., Regnell, B. and Wholin, C. Using Students as Subjects – A comparative Study of Students & Professionals in Lead-Time Impact Assessment". *4th Conference on Empirical Assessment & Evaluation in Software Engineering (EASE)*, Keele University, UK, pp. 201-214, 2000.
- [10] Höst, M. Introducing Empirical Software Engineering Methods in Education. *Proceedings of the 15th Conference in Software Education and Training (CSEET'02)*, pp. 170-179, 2002.
- [11] Juristo, N. and Moreno, A. *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers, 2001.
- [12] Kitchenham, B., Pfleger, P., Hoaglin, D., Rosenberg, J. Preliminary Empirical Research in *IEEE Transactions on Software Engineering*, Vol. 28 N° 8, pp. 721-727.
- [13] Perry, D., Porter, A. and Finkelstein, A. Studies of Software Engineering. *Future of Software Engineering*, Anthony Finkelstein, 2002.
- [14] Serrano, M., Calero, C. Empirical Study with I Methods, 1st International Empirical Studies in Software Engineering. Bunse, C., Jedlitschka, Finland, 2002.
- [15] Singer, J. and Vinson, J. Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering*, Vol. 28 N° 12, pp. 1171-1177.
- [16] Software Process Engineering Specification; adopted 1.0. Object Management Group (2002). Available in <http://www.omg.org/bin/doc?ptc/02-05-03>.
- [17] Wohlin, C., Rumsen, P., Regnell, B. and Wesslé, L. *Empirical Software Engineering*. Kluwer Academic Publishers, 2001.