

Enseñanza y Aprendizaje de Ingeniería de Computadores

**Revista de Experiencias
Docentes en Ingeniería de
Computadores**

Número 3, Mayo 2013



Edita: Departamento de
Arquitectura y Tecnología de
Computadores



Colabora: Vicerrectorado para la
Garantía de la Calidad

ENSEÑANZA Y APRENDIZAJE DE INGENIERÍA DE COMPUTADORES
Revista de Experiencias Docentes en Ingeniería de Computadores

TEACHING AND LEARNING COMPUTER ENGINEERING
Journal of Educational Experiences on Computer Engineering

Número 3, Año 2013

Comité Editorial:

Miembros de la Comisión Docente del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada:

Mancia Anguita López	Alberto Guillén Perales
José Luis Bernier Villamor	Luis Javier Herrera Maldonado
Pedro A. Castillo Valdivieso	Gonzalo Olivares Ruiz
Miguel Damas Hermoso	Julio Ortega Lopera
Javier Diaz Alonso	Begoña del Pino Prieto
Antonio Díaz García	Beatriz Prieto Campos
F. Javier Fernández Baldomero	Alberto Prieto Espinosa
Francisco Gómez Mula	Manuel Rodríguez Álvarez
Jesús González Peñalver	Fernando Rojas Ruiz

Colaboradores externos de otras Universidades:

José Manuel García Carrasco (Universidad Murcia)
Consolación Gil Montoya (Universidad de Albacete)
Diego R. Llanos Ferraris (Universidad de Valladolid)
Francisco J. Quiles Flor (Universidad de Castilla-La Mancha)
Manuel Ujaldón Martínez (Universidad de Málaga)
Miguel Ángel Vega Rodríguez (Universidad de Extremadura)

ISSN: 2173-8688 **Depósito Legal:** GR-899/2011

Edita: Departamento de Arquitectura y Tecnología de Computadores

Imprime: Copicentro Editorial

© Se pueden copiar, distribuir y comunicar públicamente contenidos de esta publicación bajo las condiciones siguientes (<http://creativecommons.org/licenses/by-nc-nd/3.0/es/>):

Reconocimiento – Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciator (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).

No comercial – No puede utilizar esta obra para fines comerciales.

Sin obras derivadas – No se puede alterar, transformar o generar una obra derivada a partir de esta obra.

Printed in Spain

Impresa en España

Índice

Editorial	1
Motivación para la Ingeniería de Computadores <i>J. Ortega, M. Anguita, M. Damas, J. González:</i>	3
Aprendizaje Cooperativo y Portafolio Digital de Grupo: Desarrollo de Competencias <i>C. Gil, R. Baños, M.G. Montoya, R.I. Herrada, F.G. Montoya:</i>	23
Plan de Acción Tutorial para Alumno de Planes de Estudios a Extinguir en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación (TIPAT) <i>M. Rodríguez, M.G. Arenas, P. Paderewski, E.M. Ramos, J. García, M.A. Rubio, N. Rico, P.A. Castillo</i>	39
Estudio sobre algoritmos genéticos en la nube y el modelo de programación MapReduce <i>G. Muñoz, P. García, P.A. Castillo, M G. Arenas, A.M. Mora, J.J. Merelo</i>	49
Estrategia de Aprendizaje Activo y Cooperativo para Periféricos y Dispositivos de Interfaz Humana en el Grado de Ingeniería Informática <i>F. Rojas</i>	69
Propuesta docente para la nueva asignatura de Tecnologías Emergentes en el Grado en Ingeniería Informática <i>S. Romero</i>	77
Implementación de Algoritmos en Hardware: de la puerta NAND al bucle FOR <i>J. Díaz</i>	95
Complementos para Informática Industrial del perfil de Ingeniería de Computadores del Grado de Informática de la UGR: Controladores Lógicos Programables <i>M. Damas, O. Baños, G. Olivares, F. Gómez</i>	107
La asignatura Informática Industrial y la función del Ingeniero Informático en la Industria <i>G. Olivares, M. Damas, F. Gómez</i>	121
Servidores Web de Altas Prestaciones <i>P.A. Castillo, J.L. Bernier, A.M. Mora, P. García, M G. Arenas</i>	133
Instrucciones para autores	143

Editorial

Este tercer número de la revista Enseñanza y Aprendizaje de Ingeniería de Computadores aparece coincidiendo con el final del periodo lectivo del curso 2012/2013, en el que por primera vez se han impartido asignaturas específicas de la especialidad de Ingeniería de Computadores, en el nuevo grado en Ingeniería Informática de la Universidad de Granada. De hecho, en Diciembre se celebraron las III Jornadas de Coordinación Docente y de Empresas, JCDE 2012, (<http://atccongresos.ugr.es/jcde/>) en las que, en esta ocasión, se han analizado las asignaturas optativas a impartir por el Departamento de Arquitectura y Tecnología de Computadores en las distintas especialidades del grado en Ingeniería Informática de la Universidad de Granada, y han tenido lugar presentaciones de las empresas Lacroix Sofrel y Seven Solutions, que han mostrado distintas perspectivas profesionales para los Ingenieros Informáticos con la especialidad de Ingeniería de Computadores. Por tanto, en estas tres primeras ediciones de las JCDE se han presentado las asignaturas que el Departamento imparte en los nuevos Grados de Ingeniería Informática y Telecomunicaciones, y han participado las empresas relacionadas con la Ingeniería de Computadores con las que el Departamento colabora. En próximas ediciones se abordarán nuevos temas relacionados con la coordinación y la evolución de la docencia de las asignaturas que el proceso de Bolonia vaya determinando.

La mayoría de los trabajos que se publican en este número amplían la descripción de los contenidos, la metodología y demás aspectos docentes de las asignaturas que se han presentado en las JCDE 2012. Así, Fernando Rojas describe la perspectiva desde la que se imparte “Periféricos y Dispositivos de Interfaz Humana”, y Samuel Romero aborda la necesidad de la formación continua y el conocimiento de unas tecnologías en constante evolución, y muestra cómo incluir estas cuestiones en la asignatura “Tecnologías Emergentes”. En el artículo de Javier Díaz Alonso se analizan cuestiones relativas a los niveles de abstracción en el diseño hardware y se describen los contenidos y la metodología para “Implementación Hardware de Algoritmos”. En los trabajos de Miguel Damas y colaboradores y Gonzalo Olivares y colaboradores, se analiza el papel del ingeniero informático en la industria a través de la descripción de la metodología y los contenidos de dos asignaturas que conforman los complementos de informática industrial para el Grado de Ingeniería Informática, “Controladores Lógicos Programables” e “Informática Industrial”. Para finalizar este bloque de artículos dedicados a la docencia de asignaturas, en la descripción de la asignatura “Servidores Web de Altas Prestaciones”, Pedro A. Castillo y colaboradores ponen de manifiesto que el rendimiento de un sitio web eficiente implica tanto aspectos relacionados con la selección de la plataforma hardware como los que afectan a la propia organización del sitio y a la implementación de las aplicaciones.

Junto a los mencionados seis artículos, en el presente número se incluyen cuatro colaboraciones más. En “Motivación para la Ingeniería de Computadores”, Julio Ortega y colaboradores abordan las cuestiones relacionadas con la perspectiva desde la que los estudiantes contemplan los estudios relacionados con los niveles más próximos al hardware, se analizan las causas, y se proponen algunas estrategias docentes para mejorar dicha perspectiva. El uso del aprendizaje cooperativo y del portafolio digital de grupo en la enseñanza de Arquitectura de Computadores se aborda en el artículo de Consolación Gil y colaboradores. La descripción y las experiencias de aplicación del plan de tutorías para estudiantes de estudios a extinguir en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación se muestra en el artículo de Manuel Rodríguez Álvarez y colaboradores. Para completar esta presentación de contenidos, en “Estudio sobre algoritmos genéticos en la nube y el modelo de programación MapReduce”, G. Muñoz y colaboradores, describen el desarrollo de un proyecto de fin de carrera en Ingeniería Informática centrado en paradigmas recientes con grandes posibilidades de desarrollo profesional como son el *cloud computing* y el modelo *MapReduce*.

Por otra parte, con este número de la revista se ha iniciado la incorporación a su Comité Editorial de profesores de otras Universidades españolas. Queremos agradecer la buena disposición de todos los profesores con los que se ha contactado hasta el momento, y que han aceptado la propuesta. Proseguiremos invitando a participar en esta experiencia a más profesores de otras Universidades con el propósito de alcanzar una mayor difusión de la revista en cuanto a la participación de autores de otras Universidades involucrados en la docencia de contenidos de Ingeniería de Computadores.

Como en anteriores números de la revista, terminamos esta presentación/editorial expresando nuestro agradecimiento a todos los que han contribuido a la celebración de las JDCE, en este caso de su edición de 2012, especialmente a Joaquín Fernández Valdivia, director de la E.T.S.I.I.T. por su participación en el acto de inauguración de la JDCE; a los ponentes en las mesas redondas y a los representantes de las empresas Lacroix Sofrel y Seven Solutions. A todos ellos, muchas gracias por su participación y su trabajo. También agradecemos al Departamento de Arquitectura y Tecnología de Computadores, y especialmente al Vicerrectorado de Garantía de la Calidad de la Universidad de Granada el apoyo y la financiación recibida.

El Comité Editorial

Motivación para la Ingeniería de Computadores

Julio Ortega, Mancia Anguita, Miguel Damas, Jesús González
Departamento de Arquitectura y Tecnología de Computadores
E.T.S.I.I.T., Universidad de Granada
 {jortega,manguita,mdamas,jesusgonzalez}@ugr.es

Resumen. Los estudios de Ingeniería de Computadores proporcionan competencias que permiten participar en el desarrollo de los actuales y futuros sistemas de cómputo, con una incidencia importante en el entorno socio-económico, y por tanto con posibilidades profesionales relevantes. Este artículo incluye consideraciones y argumentos para facilitar, a los profesores involucrados en la docencia de asignaturas relacionadas con la Ingeniería de Computadores, la labor de motivación de los estudiantes. En particular se proporcionan algunas estrategias docentes para la motivación y se ilustra su uso en la docencia del lenguaje ensamblador y de los modelos de consistencia de memoria.

Palabras clave: Aplicación de las teorías sobre la motivación, Docencia de lenguaje ensamblador, Ingeniería de computadores, Modelos de consistencia de memoria, Motivación intrínseca.

Abstract. Computer engineering courses and degrees provide competencies to be involved in the development of present and future computer systems, with a relevant effect in social and economical contexts, and interesting professional opportunities for students. This paper provides some considerations and arguments to make it easier for teachers involved in subjects related with computer engineering topics to motivate their students. More specifically, the use of some motivation strategies for teaching assembly languages and memory consistency models are described.

Keywords: Applications of motivation theories, Teaching assembly language, Computer engineering, Memory consistency models, Intrinsic motivation.

1 Introducción

¿Por qué en los estudios de Ingeniería Informática, las asignaturas, intensificaciones, especialidades, etc. más próximas al hardware suelen ser menos demandadas por los estudiantes? Una de las causas posibles surge de la percepción de que, en nuestro contexto socio-económico, se necesitan menos profesionales con las competencias

asociadas a esos estudios. Por otro lado, existe una perspectiva de la Ingeniería Informática que, desde una descripción del computador como una serie de capas, considera que las competencias relacionadas con el software podrían adquirirse, incluso al nivel propio de un ingeniero, prácticamente sin conocimientos del hardware o de la arquitectura del computador.

De hecho, se puede decir que en la Informática han confluído dos corrientes, la relacionada con lo que en inglés se conoce como “computer science”, y la que tiene que ver con la denominada “computer engineering”, o incluso “electrical and electronic engineering”. Desde cierto punto de vista, el hardware se contempla como algo relacionado solo tangencialmente con la informática. Considérese como ejemplo, el uso de adjetivos tan poco científicos, como “hermosa” o “fea”, que se pueden leer en la página 5 de la traducción al castellano de la tercera edición del libro “Sistemas Operativos Modernos” de Tanenbaum [1]. Se indica allí que una de las tareas esenciales del sistema operativo es “ocultar el hardware y presentar a los programas (y a los programadores) abstracciones agradables, elegantes, simples y consistentes con las que trabajar [...] ocultando la parte fea con la parte hermosa como muestra la figura 1.2”. Sin embargo, la evolución de la tecnología de computadores ha dado lugar a una influencia cada vez más estrecha de las características del hardware y la arquitectura del computador en la programación y el desarrollo de software. Por lo tanto, se necesitan actuaciones docentes. Por otro lado, es cierto que parece haber una menor demanda de puestos de trabajo que solicitan competencias específicas propias de la especialidad de ingeniero de computadores respecto a competencias más relacionadas con el desarrollo de software. Por tanto, la justificación del interés de la Ingeniería de Computadores debe abordarse desde planteamientos que pongan de manifiesto la relevancia de los contenidos de la especialidad, no solo en el presente sino para el futuro. Es importante hacer valer las posibilidades que encierra la perspectiva de sistema que ofrece la Ingeniería de Computadores y, por supuesto, fomentar el aspecto emprendedor en los estudiantes de Ingeniería de Computadores.

En este artículo se proponen algunas estrategias docentes que, a partir de las claves que proporcionan algunas de las teorías de la motivación más relevante propuestas hasta el momento, pueden motivar el estudio de las asignaturas de tecnología, estructura y arquitectura de computadores. Si los estudiantes consideran los contenidos de estas asignaturas asequibles y útiles para su formación como Ingenieros Informáticos es más probable que la Ingeniería de Computadores sea contemplada por algunos como su profesión. Así, tras la revisión de las teorías de la motivación y del proceso de aprendizaje que proporcionamos en la Sección 2, se describen las posibilidades de la Ingeniería de Computadores para desarrollar una carrera profesional de futuro, y adquirir competencias que fomentan la iniciativa emprendedora y la innovación en el contexto de la evolución tecnológica. Para ello, en la Sección 3 se dan algunos datos acerca de las ofertas de trabajo para profesiones relacionadas con las Tecnologías de la Información y las Comunicaciones, y se analizan las posibilidades que se contemplan en un futuro tanto desde el punto de vista de las empresas como del ámbito de la investigación. Después, en la Sección 4 se describen algunas de las perspectivas de futuro para las TIC, el punto de vista desde el que la Ingeniería de Computadores contempla a la Informática, y de qué

forma puede tener un papel relevante en la evolución hacia las nuevas aplicaciones y posibilidades previamente descritas. En la Sección 5 se proporcionan algunas estrategias docentes que pueden plantearse a partir de las teorías del aprendizaje y la motivación y de lo expuesto en las secciones anteriores. También se describe en esta sección la aplicación de esas estrategias en algunos contenidos de asignaturas de Estructura y Arquitectura de Computadores impartidas en el Grado de Ingeniería Informática de la Universidad de Granada y cursadas por los estudiantes antes de la elección de especialidad. Finalmente, la Sección 6 recoge las conclusiones del artículo.

2 El aprendizaje y la motivación

Para proponer estrategias docentes que promuevan la demanda de los estudios de Ingeniería de Computadores (Sección 5) se han tenido en cuenta las conclusiones de una serie de trabajos, recientes y/o ampliamente citados, acerca de cómo y en qué circunstancias el cerebro absorbe y almacena información, incidiendo por tanto positivamente sobre el aprendizaje [2, 3]. Estas conclusiones se pueden resumir a través de siete principios:

1. Cuando inician el estudio de una asignatura, los estudiantes parten de ciertas concepciones previas de lo que les rodea, incluyendo los contenidos de la asignatura. Las ideas de partida sobre la asignatura afectan (facilitando o dificultando) a su aprendizaje.
2. Para asimilar nuevas competencias no es suficiente conocer muchos datos sino que hace falta incorporarlos dentro de un marco conceptual que organice e interconecte los hechos para que sea fácil acceder a ellos y aplicarlos. La forma según la que los estudiantes organizan el conocimiento influye en lo que aprenden y en la forma en que lo utilizan.
3. La motivación determina, orienta y mantiene el aprendizaje de los estudiantes.
4. El conocimiento propio de los expertos se consigue integrando y practicando con los elementos que constituyen el campo de trabajo de forma que se adquieren habilidades y se sabe cuando aplicar lo aprendido.
5. Las actividades prácticas orientadas a alcanzar objetivos, junto con la realimentación acerca de lo que se ha alcanzado incrementa la calidad del aprendizaje.
6. El aprendizaje se ve afectado por el clima social, emocional, e intelectual de la clase.
7. Para conseguir el aprendizaje autodirigido, los estudiantes deben ser capaces de estimar la dificultad de la tarea en cuestión, evaluar sus propias habilidades y conocimientos, planificar su trabajo, analizar su progreso y ajustar sus estrategias convenientemente. Es conveniente proporcionar información “metacognitiva” que permita al estudiante ser consciente de su aprendizaje y del logro de las metas establecidas.

Según lo anterior, es importante tener en cuenta la visión previa que los estudiantes tienen de las asignaturas, relacionadas con la Ingeniería de Computadores, que preceden a la elección de especialidad (en el Grado en Ingeniería de Computadores de la Universidad de Granada serían Tecnología y Organización de Computadores, Estructura de Computadores, Arquitectura de Computadores, e Ingeniería de Servidores). En estas asignaturas no se estudia el computador buscando únicamente alcanzar competencias relacionadas con el diseño hardware, sino que se aportan conceptos esenciales para entender el funcionamiento del computador, de los sistemas operativos, y del desarrollo de programas eficientes.

El tercero de los siete principios indicados hace referencia a la importancia de la motivación en el aprendizaje. Dado que la motivación es un estado interno que promueve y controla la actividad de un individuo hacia un objetivo, motivar a alguien es estimularlo para que encamine sus actividades hacia ese objetivo. A continuación se resumen las características fundamentales de algunas de las teorías de la motivación más relevantes [4]: la teoría de los objetivos, la de las atribuciones, la de expectativa-valor, y la de la autodeterminación. Como se pone de manifiesto a través de dichas teorías, en la motivación influyen muchos factores.

La teoría de los objetivos relaciona la motivación con las características de los objetivos planteados. Así, dichos objetivos se califican de acuerdo con su *naturaleza* (se busca ser competente en algo o ser el que mejor calificaciones obtiene) o con su *orientación* (se busca alcanzar algo o se pretende evitar algo que no se desea). Combinando las dos posibilidades de los objetivos en cuanto a naturaleza con las dos orientaciones se tienen cuatro alternativas de motivación de la actividad. No todas esas alternativas son igual de efectivas para el aprendizaje. Así, los objetivos que pretenden alcanzar destrezas o dominar tareas producen, con mayor probabilidad, un aprendizaje más profundo que si buscan a obtener buenas calificaciones. Por otra parte, también es mejor si la acción se dirige a alcanzar algo que a huir de algo. Es decir que, según esta teoría, motiva más el deseo de adquirir una habilidad que el de evitar un suspenso.

La teoría de las atribuciones propone que la motivación de las personas está relacionada con sus expectativas de alcanzar un objetivo, ya que si alguien cree que no podrá conseguir algo, muy posiblemente no intentará alcanzarlo. Las causas del éxito o del fracaso se pueden deber a razones internas de la persona o externas a ella, que pueden cambiar con el tiempo o se mantienen invariables (dinámicas o permanentes), y son o no son controlables por la persona. Por ejemplo, mientras que el esfuerzo se puede catalogar como una causa interna, variable, y controlable; la dificultad se contempla como una causa externa, permanente y no controlable; y la suerte como externa, inestable, y no controlable. Es más probable que un estudiante persevere en una actividad y alcance el éxito en una tarea si se evita que dicho éxito se relacione con causas externas, permanentes, o incontrolables.

La teoría de expectativa-valor establece que la motivación hacia algo aumenta cuando existe la creencia de que se puede alcanzar el objetivo en cuestión (expectativa) y merece la pena hacerlo (valor). En cuanto a las expectativas de éxito en los objetivos,

éstas dependen de la percepción que se tiene acerca de la dificultad que entraña alcanzarlos, y de la capacidad que uno considera que tiene para ello. También relacionados con esta teoría, en [5] se describen tres factores importantes que determinan la motivación de las personas hacia algo: (a) la forma en que cada uno contempla sus habilidades o capacidades para hacer ese algo; (b) lo que se cree importante, interesante o útil; y (c) los sentimientos y emociones que suscita el objetivo a alcanzar. Por tanto, desde este punto de vista, la motivación se consigue poniendo de manifiesto la relevancia de los objetivos a alcanzar y creando entornos docentes en los que los estudiantes tengan la sensación de que pueden completar las tareas que se les planteen.

La teoría de la autodeterminación distingue entre dos tipos de motivación. La motivación *extrínseca*, cuando el estímulo para la acción proviene de factores externos a la persona, como por ejemplo contemplar la posibilidad de encontrar puestos de trabajo bien remunerados a la hora de elegir una determinada profesión, y la motivación *intrínseca*, en la que las razones para encaminar la actividad hacia el objetivo en cuestión están relacionadas con lo que, internamente, la persona considera interesante o importante, como por ejemplo, puede considerarse un objetivo intrínseco ser partícipe en el desarrollo de herramientas y técnicas que pueden abrir nuevos caminos y posibilidades para la humanidad. Desde hace bastante tiempo se han sucedido trabajos que buscaban las razones por las que una actividad puede ser intrínsecamente motivadora. En [6] se pueden encontrar una presentación resumida de los más relevantes, junto con las correspondientes referencias. Según algunas de las aproximaciones descritas, lo que motivaría a las personas está relacionado con el grado de control que pueden alcanzar en su relación con otras personas, con objetos, y con ellos mismos, es decir lo que se denomina *interacción efectiva*. Así, hay tres necesidades psicológicas básicas en la persona que condicionan la motivación hacia lo que las satisface [4, 7]: la autonomía (sensación de libertad y control sobre nuestra actividad); la competencia (poder ejercitar y poner de manifiesto las propias capacidades haciendo algo que se considera realmente importante para la sociedad); y la socialización (sentirse conectado socialmente). Por tanto, la relevancia de los problemas que se plantean en una disciplina y la utilidad de las soluciones constituyen la base de la motivación intrínseca hacia dicha disciplina. En trabajos heurísticos, donde interviene fundamentalmente el hemisferio derecho y en cuyo desarrollo juega un papel central la creatividad, como es el caso de las ingenierías, la motivación intrínseca juega un papel relevante frente a la motivación extrínseca [7].

A continuación, en las Secciones 3 y 4, se analizan aspectos que afectan a la motivación intrínseca y extrínseca para estudiar Ingeniería de Computadores. Así, mientras que la Sección 3 proporciona información acerca del mercado de trabajo en Informática y en Ingeniería de Computadores, como elemento importante para la motivación extrínseca, la Sección 4 proporciona algunas de las perspectivas futuras para el desarrollo profesional de un Ingeniero de Computadores, que van a ser clave para la motivación intrínseca de los estudiantes. Después, en la Sección 5 se proporcionan estrategias docentes que motiven a los estudiantes en el estudio de las asignaturas relacionadas con la Ingeniería de Computadores (las previas a la elección de la especialidad y las propias de la especialidad). Para establecer esas estrategias se

ha tenido en cuenta las conclusiones acerca de la relevancia de los contenidos y los entornos docentes de la teoría de expectativa-valor.

3 La demanda de Ingenieros de Computadores

En el Computer Curricula de 2005 de ACM e IEEE [8] se establecen cinco perfiles o disciplinas de computación: Ingeniería de Computadores, Ingeniería del Software, Tecnología de la Información, Sistemas de Información, y Ciencia de la Computación. Para establecerlos se tuvo en cuenta la existencia de guías curriculares publicadas para el perfil correspondiente por una o más sociedades científicas o profesionales. Estos perfiles coinciden (salvo por el uso del plural en el perfil de Tecnología de la Información y por el añadido de Sistemas Inteligentes y el uso de Computación en lugar de Ciencia de la Computación) con las especialidades que contempla el título de Grado en Ingeniería Informática de la Universidad de Granada (aprobado oficialmente en 2010). Una descripción detallada de las especialidades propuestas por IEEE/ACM se tiene en [8], en tanto que las características de las especialidades en el Grado en Informática de la Universidad de Granada están en [9].

En [10, 11] se pueden consultar distintos análisis del sector TIC en Andalucía elaborados por SANDETEL (Sociedad Andaluza para el Desarrollo de las Telecomunicaciones). Según el informe del año 2010 [10], un 83.5% de los 32.000 puestos de trabajo del sector TIC correspondía a empleados con formación superior, y comprendía más de 1600 empresas (un 10% de esas empresas eran granadinas, lejos del 36% y del 27% correspondientes a Sevilla y Málaga, respectivamente), con una facturación de 4200 millones de euros. El 59% de las empresas realizaban actividades de I+D+i, cuya financiación promedio había sido del orden del 16.6% de la facturación, y provenía en un 68% de la propia empresa, y en un 25% de fuentes públicas. Teniendo en cuenta la distribución de actividades que hace la CNAE (Clasificación Nacional de Actividades Económicas), alrededor del 69% de las empresas se dedicaban a la “programación, consultoría y otras actividades relacionadas con la informática”, el 46% ofrecían “servicios de portales web o proceso de datos”, y el 17% se dedicaba a la “fabricación de productos informáticos, electrónicos y ópticos”. En [9] se proporciona una descripción análoga para toda España.

Es posible tener una referencia bastante completa de la demanda de profesionales en tecnologías de la información y las comunicaciones en EE.UU. a través de la página *Bureau of Labor Statistics* (<http://www.bls.gov/ooh/home.htm>). En la Tabla 1 se muestran las ofertas de trabajo y los salarios anuales en 2010 para las profesiones que se recogen en dicha página dentro del campo Tecnologías de la Información y las Comunicaciones, más el campo Ingeniería de Hardware de Computador. En particular, en este último campo se especifica que la mayoría de los empleados tienen un grado en Ingeniería de Computadores. Se puede ver que profesiones con menos ofertas de trabajo suelen tener salarios mayores

Una primera cuestión que plantea la distribución de actividades de la CNAE utilizada en [10, 12], es que es difícil distinguir demandas de profesionales que correspondan a las competencias específicas de cada una de las distintas especialidades ofertadas en el grado de Ingeniería Informática de la Universidad de Granada, bastante usuales, por otra parte, en los grados de Ingeniería Informática de otras Universidades (incluso existen como titulaciones de grado en alguna de ellas): Ingeniería del Software, Computación y Sistemas Inteligentes, Sistemas de Información, Tecnologías de la Información, e Ingeniería de Computadores. Incluso en el caso de las profesiones que considera el *Bureau of Labor Statistics* en EE.UU., no existe una correspondencia clara (salvo quizá en el caso de la Ingeniería de Computadores). Así, si consideramos la clasificación de la CNAE y centrándonos en las competencias de un Ingeniero Informático con la especialidad de Ingeniería de Computadores [9], éstas parecerían más directamente relacionadas con las del apartado de “fabricación de productos informáticos, electrónicos y ópticos”, pero también tendrían cabida en todas aquellas empresas que requieren desarrollar software para productos embebidos, desarrollar software eficiente, configurar y mantener equipos informáticos, o las que proporcionan el acceso a dichos recursos desde el punto de vista del paradigma actual del *cloud* [13].

Tabla 1. Salarios y ofertas de trabajo en TIC en 2010 según el *Bureau of Labor Statistics* (EE.UU.)

Profesión	Salario Anual (2010) (\$)	Ofertas de trabajo (2010)
Computer and Information Research Scientists	100.660	28.200
Computer Support Specialists	46.260	607.100
Software Developers	90.530	913.100
Information Security Analysts, Web Developers, and Computer Network Architects	75.760	302.300
Database Administrators	73.490	110.800
Computer Programmers	71.380	363.100
Computer Systems Analysts	77.740	544.400
Network and Computer Systems Administrators	69.160	347.200
Computer Hardware Engineering	98.810	70.000

Aparentemente, las consideraciones científicas acerca de los ámbitos de actividad del Ingeniero Informático no tienen una relación demasiado estrecha con las actividades que se recogen en la CNAE, poniéndose una vez más de manifiesto la necesidad de un contacto más estrecho entre el mundo universitario y el empresarial. Si cabe, dado que una gran mayoría de ofertas de trabajo buscan perfiles profesionales relacionados con la programación, en muchos casos se suelen identificar las especialidades de Ingeniería Informática que tienen que ver con desarrollo de software, sistemas de información, e incluso tecnologías de la información como las más prometedoras desde el punto de vista de las oportunidades laborales. La especialidad de Ingeniería de Computadores, se contempla como la menos relacionada con el software del computador. En cualquier caso, cuando se trata de salir de una situación de crisis económica, o se pretende impulsar un sector manifiestamente mejorable, es más

importante analizar las expectativas de futuro que puede ofrecer una titulación determinada que contabilizar las limitadas posibilidades que existen. Por lo tanto, desde el punto de vista de este artículo, lo que habría que plantearse es si tal o cual especialidad permite adquirir unas competencias próximas, no ya a las que se demandan actualmente, sino a las que se demandarán en un futuro próximo. Más que describir la situación actual respecto a un tipo determinado de profesionales, lo importante es analizar sus perspectivas de futuro.

Así, aunque es difícil establecer una predicción suficientemente precisa del mercado de trabajo para los Ingenieros de Computadores en el futuro, se pueden identificar las opciones más probables, los yacimientos donde se puede encontrar materia prima para la creatividad, la innovación, y para el emprendimiento por parte de Ingenieros Informáticos. Para ello hemos utilizado dos tipos de fuentes. Por un lado, se han analizado documentos de organismos públicos o asociaciones de empresas nacionales como SANDETEL y AMETIC (Asociación Multisectorial de Empresas de Electrónica, Tecnologías de la Información, Telecomunicaciones, y Contenidos Digitales, www.ametic.es). Por otro lado, se han tenido en cuenta los análisis realizados por redes de excelencia como HIPEAC (High Performance and Embedded Architecture and Compilation, www.hipeac.net), relacionadas con el ámbito de la Ingeniería de Computadores. Obviamente, no se trata de hacer aquí una presentación exhaustiva de los documentos a los que nos referiremos sino simplemente de proporcionar esas referencias para su consulta, y de extraer algunas conclusiones que consideramos interesantes desde el punto de vista de la motivación de nuestros estudiantes.

En [14] se indica que: “Las tecnologías de la información y las comunicaciones (TIC) se plantean el reto de ofrecer servicios, sistemas y aplicaciones respondiendo al paradigma *siempre conectado y en cualquier lugar*. Este paradigma se ha convertido en un objetivo que impulsa la innovación y que está promoviendo que la industria de los servicios y contenidos electrónicos se mueva a gran velocidad.” En [15] también se hace hincapié en las expectativas empresariales que surgen del aprovechamiento de los dispositivos móviles y de la accesibilidad a toda la tecnología relacionada, y se sitúan las posibilidades que se abren con el acceso móvil a Internet al nivel de lo que supuso la aparición del computador personal o del propio Internet. En cualquier caso, sectores de nuestra economía como la salud, el transporte, el turismo, etc., van a encontrar nuevas posibilidades de crecimiento en el acceso, a través de dispositivos móviles, a datos y servicios promoviendo además las infraestructuras de *cloud*.

Existe una coincidencia entre el planteamiento previo, que parte de las empresas, y el punto de vista de la red HIPEAC, que puede considerarse representativo del ámbito académico y científico. Así, en [16] se defiende que el futuro inmediato de los computadores vendrá determinado por la programación eficiente de arquitecturas paralelas heterogéneas, el diseño de infraestructuras que escalen adecuadamente al aumentar la complejidad y el volumen esperado de datos, y la computación ubicua fiable y predecible. Según esto, las áreas de trabajo y los objetivos del ingeniero de computadores estarían relacionados con:

- *La eficiencia de los sistemas.* Maximizar la computación por unidad de energía de las arquitecturas, explorando el diseño de arquitecturas heterogéneas y la gestión de la localidad y las comunicaciones.

- *La gestión de sistemas cada vez más complejos.* Proporcionar herramientas que permitan el desarrollo de software para las nuevas arquitecturas multi-núcleo heterogéneas, las nuevas generaciones de núcleos de procesamiento, y la optimización integral de los diseños (optimización inter-componente e inter-layer).

- *La mejora de la fiabilidad de los sistemas y las aplicaciones.* Tanto el desarrollo de arquitecturas para las aplicaciones de procesamiento intensivo en datos (*Data Deluge*), como el de plataformas fiables para la computación ubicua.

Además, estas áreas y objetivos tendrán incidencia en los sectores que se consideran estratégicos en la Unión Europea. Concretamente, los nuevos sistemas de cómputo ocasionarán cambios relevantes en el sector energético, el transporte y la movilidad, la salud, el medio ambiente, el cuidado de los mayores, la productividad, la fiabilidad, la seguridad, y la educación.

4 La Ingeniería de Computadores y la Informática

Muchas de las oportunidades que las tecnologías de la información y las comunicaciones ofrecen para contribuir a la mejora de la productividad y al progreso social, y que permiten desarrollar una carrera profesional creativa y con futuro, requieren competencias propias de la Ingeniería de Computadores. En esta sección analizaremos lo que aportan al ingeniero informático los conocimientos desde los que el ingeniero de computadores contempla los sistemas de cómputo. Así, proporcionaremos algunos ejemplos de la influencia de la arquitectura de los sistemas (y los niveles que se suelen identificar como hardware) en la programación y en las características de los lenguajes de programación, dado que con frecuencia se olvida la importancia de los niveles próximos al hardware en el desarrollo de proyectos informáticos relevantes, y se suele identificar el papel del ingeniero informático con el del desarrollo de software usando lenguajes de programación de alto nivel que permiten “aislarse” de las características del hardware.

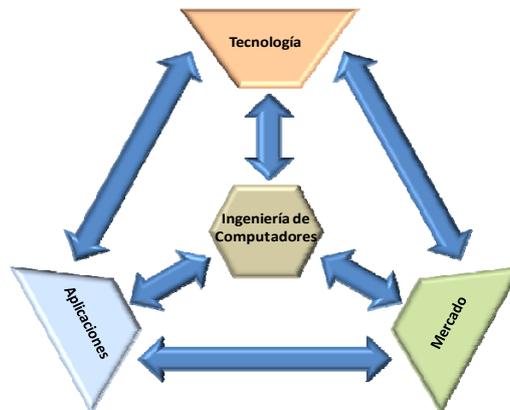


Figura 1. Interacción Tecnología-Aplicaciones-Mercado-Ingeniería de Computadores

La ingeniería de computadores aborda la informática desde un punto de vista sistémico. Integra los aspectos científicos y tecnológicos del diseño, implementación y mantenimiento de los componentes hardware y software de los sistemas de cómputo, las redes, y los equipos controlados por computador, que se utilizan en dominios de aplicación muy diversos [17] donde se necesitan conocimientos de las características hardware de la plataforma y del desarrollo de un software que aproveche eficientemente las posibilidades de la misma.

La evolución científica y tecnológica, según la teoría de las revoluciones científicas de K. S. Kuhn, se produce a través de paradigmas que surgen para dar respuestas a nuevos problemas y deficiencias de paradigmas previamente establecidos. Tal y como muestra la Figura 1, en la Ingeniería de Computadores los paradigmas evolucionan a partir de la interacción entre las posibilidades que proporciona la tecnología y las características de las aplicaciones que demanda el *mercado*, definido por los factores económicos (oferta-demanda) que introducen condicionantes en las aplicaciones y en la viabilidad de las tecnologías y los sistemas de cómputo. Esta interacción entre tecnología, aplicaciones y mercado determina las perspectivas profesionales futuras de los ingenieros de computadores al definir los problemas a los que las nuevas plataformas de cómputo deben dar respuestas. Según el paradigma actualmente vigente, estamos rodeados de dispositivos y computadores, interconectados, y con capacidades de procesamiento cada vez mayores. De hecho, en todas las actividades se utilizan artefactos, aparatos, y utensilios que pueden mejorarse en cuanto a sus funciones, a su automatización, y a su control. La tecnología ha permitido la sustitución de elementos de control basados en electrónica analógica por sistemas digitales basados en microcontroladores y procesadores cada vez más potentes, de menor tamaño y menor consumo, y su uso en aplicaciones donde antes no podía pensarse en que se podrían utilizar por precio, dimensiones, consumo, o falta de prestaciones.

Las oportunidades que se pueden generar sólo van a estar limitadas por el ingenio, y las capacidades de emprendimiento y financiación. Hay que recordar que la población actual del mundo es de alrededor de 7×10^9 personas y hay 4.3×10^9 direcciones IP (con IPv6 hay 3.4×10^{38} direcciones posibles), aunque se estima que para 2020 habrá alrededor de 50×10^9 dispositivos conectados a la red (6.58 dispositivos por persona), gran parte de ellos integrados en objetos cotidianos, como los *smartphones*, los automóviles, los electrodomésticos, o la ropa. Esta tendencia ha dado lugar a un ritmo de crecimiento considerable en el consumo de sistemas empujados y procesadores para teléfonos móviles, en comparación con el del consumo de procesadores para lo que se reconoce por el gran público como “computador” (servidores, PCs y portátiles). Por tanto, en los próximos años surgirá una gran diversidad de aplicaciones englobadas en lo que se *denomina Internet de las Cosas*. De hecho, el volumen de mercado de los denominados *smart systems*, incluyendo aplicaciones en los sectores de la seguridad, la salud, el transporte, la educación, la automatización industrial, la energía, y el hogar, se estima en más de un billón (10^{12}) de dólares para 2016 (www.marketsandmarkets.com/). Teniendo en cuenta el volumen de dispositivos y sistemas de cómputo interconectados, se estima que en 2017 el tráfico de datos será superior a los 11.2 exabytes por mes [18]. Por otra parte, la información almacenada (y replicada) a finales de 2011 era de 1.8 Zettabytes (1.8×10^{21} bytes = 1.8 billones de Gbytes), y aumenta el número de aplicaciones y actividades empresariales que tienen que ver con el acceso a bases de datos cada vez mayores. Las necesidades de almacenamiento, acceso, y procesamiento de esos datos también promoverán un número considerable de innovaciones y mercados, que mantendrán el interés por la supercomputación y marcarán la evolución del *cloud computing* [14, 19] y los *centros de datos*. La relevancia de la supercomputación en los ámbitos económico y social, va a mantener el interés por superar los considerables retos tecnológicos que plantea la constante búsqueda de plataformas cada vez más potentes. Así, en la lista de Noviembre de 2012 del TOP500 (www.top500.org) se superaron los 17 PFLOPS (17×10^{15} operaciones en coma flotante por segundo), tan sólo cuatro años después de alcanzar el PFLOPS en 2008, pero la necesidad de más velocidad de cómputo se mantiene. En este ámbito, también es clave el consumo de energía (hay estudios que indican que en EE.UU. las TIC están en el segundo lugar como sector consumidor de energía), como pone de manifiesto la lista GREEN500 (www.green500.org). Al evaluar las prestaciones de los supercomputadores importa la velocidad, pero también la eficiencia energética.

Se puede considerar que la Ingeniería de Computadores proporciona el eslabón entre las prestaciones y el funcionamiento correcto del sistema. Cuando además del resultado importan las restricciones de tiempo establecidas para obtenerlo y/o el consumo energético, los conceptos que aborda la Ingeniería de Computadores son esenciales. Por ejemplo, la necesidad de obtener cada vez más velocidad de procesamiento con un consumo energético limitado ha orientado la evolución de los microprocesadores hacia microarquitecturas que integran varios núcleos de procesamiento junto con memorias cache, controladores de memoria, interfaces con buses de E/S e interconexiones, etc. Estas microarquitecturas multi-núcleo están presentes en prácticamente todos ámbitos de aplicación de los dispositivos de cómputo. Por ejemplo, se pueden usar varios núcleos para mantener el nivel de servicio con menos consumo de energía: los teléfonos móviles con arquitecturas

multi-núcleo podrían trabajar a menor frecuencia y consumir menos batería. El dominio de conceptos de concurrencia y paralelismo es una necesidad cada vez más acuciante. Para los programadores, la mejora de las prestaciones ha dejado de estar servida en bandeja sin esfuerzo con cada nueva generación de microprocesadores. Ahora hay que escribir software que extraiga explícitamente el paralelismo que ofrecen las arquitecturas multi-núcleo (en muchos casos heterogéneas) para cuya programación no existen herramientas totalmente adecuadas hasta el momento. Los códigos que no sean capaces de aprovechar la presencia de varios núcleos de procesamiento, se encontrarán en una clara desventaja competitiva.

En [20,21] se pone de manifiesto cómo las características de las arquitecturas multi-núcleo pueden hacerse visibles para el programador de alto nivel. Por ejemplo, en una arquitectura multi-núcleo la computación puede ser bastante menos costosa que la comunicación entre núcleos (aunque los anchos de banda entre núcleos pueden ser del orden de los 3 GB/s, las latencias pueden llegar casi al centenar de ciclos), o un acceso a la memoria compartida, que debe mantener la coherencia. En las arquitecturas secuenciales usualmente se almacenan los datos que se van obteniendo para evitar tener que recalcularlos, dado que esto suele costar más tiempo que acceder a registros locales donde se pueden almacenar los resultados. En arquitecturas con memoria físicamente distribuida entre los procesadores, no está tan claro que el acceso de un núcleo a un dato calculado por otro núcleo sea más rápido que volverlo a calcular. Sería deseable que los lenguajes de programación permitiesen diseñar programas en los que la decisión entre almacenar o recalcular se pudiera retrasar más allá del momento de la compilación. Con esto se podría mejorar la portabilidad de los programas entre distintas arquitecturas.

Otra muestra de la influencia de la arquitectura en los lenguajes de programación se encuentra en el paso de parámetros, por referencia o valor, en las llamadas a las funciones. La opción de utilizar el paso por referencia, usualmente preferida en una arquitectura con un solo núcleo, puede no ser la alternativa más eficiente en una arquitectura multinúcleo, donde pueden producirse penalizaciones considerables si el dato al que se hace referencia está en memoria compartida y se tiene un número elevado de núcleos. Según esto, y en la misma línea de lo indicado en relación con las alternativas de almacenar o recalcular, sería conveniente que los lenguajes permitieran la decisión retrasada del procedimiento para el paso de parámetros (late-binding) según las características de la arquitectura.

Estos ejemplos ponen de manifiesto que la tendencia actual en el diseño de nuevas arquitecturas hace que se necesiten lenguajes de programación que hagan énfasis en la concurrencia y permitan que, para desarrollar códigos eficientes de forma natural, los programadores tengan presente ciertas características de las arquitecturas. El ingeniero informático que tenga una mayor comprensión de estas situaciones no solo podrá desarrollar proyectos más competitivos sino que, además, tendrá más posibilidades de proponer innovaciones en el diseño de nuevas herramientas y plataformas. En este contexto, la especialidad de Ingeniería de Computadores contribuye, además, a desarrollar en el estudiante la perspectiva del aprovechamiento eficiente de los sistemas.

5 Estrategias docentes para la motivación

En esta sección se proponen algunas estrategias de enseñanza/aprendizaje para promover la demanda de asignaturas relacionadas con la Ingeniería de Computadores en estudios de grado teniendo, en cuenta lo descrito en las secciones previas. Estas estrategias deben aplicarse ya en las asignaturas obligatorias relacionadas que se cursan con anterioridad al momento en que se elige una u otra especialidad. Concretamente, y como se ha indicado, en el grado en Ingeniería Informática de la Universidad de Granada, las asignaturas Tecnología y Organización de Computadores, Estructura de Computadores, Arquitectura de Computadores, e Ingeniería de Servidores son las asignaturas obligatorias que los estudiantes deben cursar antes de elegir entre las distintas especialidades (sus contenidos se pueden consultar en [9]).

Tabla 2. Estrategias docentes según las teorías de la motivación resumidas en la Sección 2

Teoría de la motivación	Estrategias docentes
Objetivos (ser competente, alcanzar los objetivos)	<ul style="list-style-type: none"> • Despertar la curiosidad planteando preguntas sobre aplicaciones prácticas de la materia que se irán respondiendo a lo largo de la asignatura • Evitar referencias que puedan convertir a las calificaciones en objetivos de aprendizaje
Atribuciones (causas internas, dinámicas, y controlables)	<ul style="list-style-type: none"> • Proponer relaciones de problemas de dificultad creciente, comenzando con problemas de resolución muy directa a partir de las leyes o expresiones matemáticas estudiadas en la asignatura • Evaluar el esfuerzo: proponer varias actividades voluntarias y tener en cuenta el número de actividades completadas.
Expectativa-Valor (metas alcanzables y relevantes)	<ul style="list-style-type: none"> • Mostrar que los conceptos que se abordan en la asignatura son importantes para ser profesionales competentes. • Realizar pruebas de evaluación sobre habilidades concretas o contenidos específicos tratados en clase con detalle.
Autodeterminación (autonomía, competencia, socialización)	<ul style="list-style-type: none"> • Plantear prácticas y ejercicios individuales que pongan de manifiesto la utilidad de los contenidos de la asignatura. • Promover el trabajo cooperativo en equipo • Describir posibilidades profesionales que abren las competencias alcanzadas con los contenidos de la asignatura

La Tabla 2 propone, en función de las distintas teorías de la motivación resumidas en la Sección 2, algunas estrategias docentes para motivar el aprendizaje en una asignatura. Las propuestas de la Tabla 2 ponen de manifiesto dos objetivos generales respecto a los contenidos de las asignaturas: (1) que sean relevantes para la profesión, y (2) que se contemplen como asequibles con un esfuerzo razonable. Precisamente uno de los 10 mandamientos de la docencia según el profesor Yale Patt [22] es no bajar el listón de los objetivos de aprendizaje. Buscar que los estudiantes asimilen y sientan que controlan el aprendizaje de una asignatura, y que son capaces de alcanzar las competencias correspondientes no debe suponer la renuncia a abordar conceptos o problemas complejos. Precisamente esos problemas son los que hacen que el estudiante llegue a ser consciente de la utilidad de la asignatura para su futuro profesional. Los que le convencen de que el esfuerzo realizado le ha permitido

avanzar en su conocimiento del computador y de aspectos de su profesión que no hubiera ni imaginado si no es por la asignatura estudiada.

Además de las teorías de la motivación, en la Tabla 2 también se han tenido en cuenta las tres categorías para los niveles de comprensión de un contenido: se conoce, se usa, y se asimila y se adapta [26]. Aunque el tercer nivel requiere un tiempo considerable (en [23] se indica que alcanzar el nivel de experto en un ámbito puede requerir del orden de 10000 horas), alcanzar los dos primeros sí que puede conseguirse en menor tiempo y pueden constituir el objetivo de una asignatura de grado de 6 créditos ECTS (150 horas). En este contexto, el modelo de enseñanza-aprendizaje en espiral [15] es muy adecuado para mejorar las expectativas de los estudiantes acerca de la consecución de los objetivos del aprendizaje, buscando la competencia y la utilidad de los conocimientos adquiridos. En dicho modelo se suceden secuencias de lección – demostración – práctica. Las clases teóricas o lecciones magistrales, sobre todo si se incluyen técnicas de aprendizaje activo que hacen posible la participación del estudiante [24], permiten transmitir conocimiento, pero el trabajo práctico es esencial para transformar el conocimiento en habilidades o competencias. Las demostraciones también son importantes para ilustrar experiencias que no pueden abordarse en prácticas, y para contrarrestar ideas preconcebidas por los estudiantes. Para ello, las herramientas de simulación, el software, y el material audiovisual accesible a través de Internet constituyen una ayuda inestimable. Por ejemplo, se puede empezar la presentación de una asignatura como Arquitectura de Computadores ejecutando diversas versiones de un mismo programa como el que se proporciona como material de apoyo en [25]. Este programa aborda una tarea de presentación de imágenes y las distintas opciones de optimización de código dan como resultado una clara mejora en la velocidad de presentación de las imágenes.

Las estrategias docentes de la Tabla 2 deben concretarse para cada asignatura describiendo su uso en los contenidos que aborda. Para completar esta sección utilizaremos como ejemplos de aplicación de esas estrategias la docencia del lenguaje ensamblador (Sección 5.1), y de la consistencia de memoria (Sección 5.2).

5.1. Enseñanza/aprendizaje del lenguaje ensamblador

El aprendizaje del lenguaje ensamblador suele concitar considerables reticencias entre los estudiantes. Respecto a esta cuestión, está ampliamente aceptado que hay que evitar que los estudiantes lo contemplen como una herramienta de programación, dado que no resiste ninguna comparación con cualquier lenguaje de alto nivel que estén utilizando para desarrollar sus programas. Es imprescindible cambiar esta perspectiva incidiendo sobre dos aspectos importantes del ensamblador. Por un lado está su faceta de lenguaje para describir lo que ocurre en la máquina al nivel adecuado para entender el efecto de su funcionamiento sobre las prestaciones del computador. Por otro lado está su utilidad a la hora de optimizar el uso que los programas hacen de la máquina. A través del lenguaje ensamblador se puede comprender cómo trabaja el procesador, cómo ve la memoria, cómo interactúan los programas con el sistema operativo, cómo se representan internamente los datos y cómo se accede a ellos, cómo se pueden optimizar los programas a alto y bajo nivel, etc. El comentario editorial del

libro [26] en la correspondiente página web de Amazon [www.amazon.com] indica que la diferencia entre un programador mediocre y un buen programador es que éste comprende el lenguaje ensamblador que, al fin y al cabo es el lenguaje del computador y permite entender como se ejecutan los programas. También se hace referencia a un comentario de Joel Spolsky [27] para el que programar sin entender como trabaja la CPU es como practicar la medicina sin saber anatomía. Si no se conoce ensamblador, no se sabe realmente lo que está ejecutando la máquina. Existen por lo tanto razones más que suficientes para convencer a los estudiantes de la importancia del ensamblador en su formación como ingenieros informáticos.

Queda el problema de abordar la enseñanza del ensamblador desde una perspectiva adecuada. Para ello, se pueden aplicar las estrategias docentes que se indican en la Tabla 2. Según ellas, habría que realizar una aproximación gradual al aprendizaje del lenguaje ensamblador en la que, por este orden: (1) se proponga la realización de programas para algoritmos sencillos de ordenación, operaciones aritméticas, etc., a través de los que se ejerciten los modos de direccionamiento, se aprendan los tipos de datos, y se tenga presente el efecto de la ubicación de los datos en la estructura del programa (en principio, no habría que incluir aspectos más complejos como el paso de parámetros); (2) se lean y analicen códigos en ensamblador generados por el compilador a partir de programas sencillos que no generen llamadas al sistema (por ejemplo bucles) y que permitirían introducir las cuestiones relativas al paso de parámetros; (3) se ensayen modificaciones en los códigos en ensamblador y se compruebe su efecto en las prestaciones del programa; y (4) se aborde finalmente algún programa más complejo en ensamblador para una aplicación más realista, donde el uso del lenguaje ensamblador proporcione alguna ventaja en cuanto a las prestaciones. Respecto a esta cuestión, la programación de algún juego sencillo suele resultar muy efectiva y motiva considerablemente a los estudiantes. Además, en las distintas asignaturas relacionadas con la Ingeniería de Computadores, el ensamblador debe utilizarse como una herramienta descriptiva más del comportamiento del computador.

5.2. Enseñanza/aprendizaje de la consistencia de memoria

La arquitectura de computadores se suele definir como la interfaz entre software y hardware. Entender la interacción del software con el hardware del computador implica tener ciertas nociones de cómo se procesan las instrucciones máquina: conocimientos del lenguaje de transferencia de registros, del aprovechamiento del paralelismo entre instrucciones, de la interacción con la memoria y los periféricos, y de sus efectos en el tiempo de procesamiento, etc., y otros contenidos que se imparten en asignaturas de Tecnología, Estructura y Arquitectura de Computadores. Otra forma de poner de manifiesto la utilidad de la arquitectura de computadores, es la optimización de código. El conocimiento de los detalles del hardware que permiten entender la forma de generar códigos más eficientes en cuanto a prestaciones y consumo, se justifica el interés de los conceptos propios de la tecnología, y la estructura y arquitectura de computadores para el desarrollo de programas eficientes.

En esta línea, el modelo de consistencia de memoria que utiliza un lenguaje de programación determina las optimizaciones que un compilador (o el programador) puede realizar manteniendo la ejecución correcta de los programas.

En esa línea, un concepto interesante para poner de manifiesto la interrelación entre el hardware y el software, de gran relevancia en el contexto actual del uso extensivo de microprocesadores multinúcleo, es el de la consistencia de memoria. El modelo de consistencia de memoria es fundamental en la semántica de un programa de memoria compartida ya que define los valores que puede devolver una instrucción de acceso a memoria (un *load*) al establecer las restricciones en el orden de acceso a memoria por parte de los distintos procesos/hebras del programa. El modelo de consistencia responde preguntas relativas a, por ejemplo, la sincronización que se necesita para asegurar que una escritura por parte de una hebra se produce antes que la lectura realizada por otra, el valor final de una posición de memoria sobre la que distintas hebras han escrito, etc.

A través del modelo de consistencia de memoria se define una interfaz entre el programa y el hardware o el software que transforma ese programa (un compilador o una máquina virtual por ejemplo) de forma que no es posible razonar sobre el programa con un modelo de consistencia ambiguo. Como se indica en [28], es difícil enseñar programación multihebra sin claridad en los modelos de memoria. Por otro lado, las características del modelo de consistencia de memoria deben procurar un equilibrio entre la concurrencia que puede aprovecharse al reducir las restricciones en el orden de los accesos a memoria, y la complejidad de la implementación y del modelo de programación que debe utilizarse. Por ejemplo, en un modelo de consistencia de memoria débil, en el que se minimizan las restricciones en el orden de los accesos de lectura o escritura a la memoria, necesita que el software proporcione información sobre los eventos de sincronización de accesos a memoria para que, en los casos en los que sea necesario, se respete cierto orden. Esta información de sincronización la proporciona bien el programador a través del correspondiente lenguaje paralelo, bien el compilador o su sistema de *runtime*. Si el software fuese capaz de proporcionar información sobre los patrones de acceso esperados para las variables compartidas, se podrían aplicar opciones de optimización que mejorasen las prestaciones del sistema. Poner de manifiesto este compromiso entre prestaciones y programabilidad, como otros compromisos que aparecen continuamente en el diseño y uso de los computadores, es relevante para la formación de un Ingeniero Informático.

La incidencia de los modelos de consistencia en el programador se puede poner de manifiesto de forma inmediata si en la asignatura donde se imparten dichos modelos se realizan prácticas con programas paralelos realizados con herramientas como OpenMP. Precisamente la sección 1.4.1 del documento “OpenMP Application Programming Interface” [29], correspondiente a las especificaciones de la versión 3.0 de OpenMP, se dedica al modelo de memoria de OpenMP y comienza indicando: “OpenMP provides a relaxed-consistency, shared-memory model”. Según esto, el programador debe saber qué son los modelos de consistencia de memoria, qué problemas que plantean, y cómo afecta al desarrollo de los programas paralelos. A

medida que se extiende el uso de los microprocesadores multinúcleo, los multiprocesadores NUMA, y los clusteres, es más importante disponer de una especificación clara del modelo de memoria y un uso correcto del mismo en los programas paralelos. En [29, 30] se pueden encontrar descripciones del modelo de consistencia de memoria OpenMP y de su relación con otros modelos de consistencia previamente propuestos. Concretamente, el modelo de memoria de OpenMP considera que existe una *memoria*, accesible a todas las hebras, para almacenar datos y acceder a ellos. Además, cada hebra dispone de una imagen de esa memoria (*memoria temporal*), que puede utilizar para almacenar datos temporalmente cuando otras hebras no necesitan acceder a ellos. Los datos pueden moverse entre *memorias temporales* y *memoria* pero para pasar de una memoria temporal a otra tienen que hacerlo a través de la memoria. Una variable utilizada en una región paralela puede ser compartida o privada. Cada referencia a una variable compartida dentro de la región paralela se refiere a la variable original del mismo nombre. En el caso de una variable privada, una referencia a dicha variable dentro de una región se refiere a la variable del mismo tipo y tamaño original pero privada para una hebra (no accesible a las demás).

Como se ha dicho, el modelo de consistencia de OpenMP es un modelo relajado (weak ordering) en el que, dentro de una hebra, se permite reordenar los accesos a variables diferentes a no ser esos accesos estén separados por una directiva FLUSH referida a ambas variables. Así, los órdenes que garantiza este modelo de consistencia son los $S \rightarrow W$, $S \rightarrow R$, $R \rightarrow S$, $W \rightarrow S$, y $S \rightarrow S$, donde S es una operación de sincronización, R es una lectura, y W una escritura. La directiva FLUSH garantiza que las operaciones de memoria previas a la directiva sobre las variables a las que se refiere dicho FLUSH deben haberse completado y tener copias en memoria antes de que acabe el FLUSH, y además, las operaciones con memoria posteriores a FLUSH no deben comenzar antes de que la directiva termine. Además, todos los valores que queden en las memorias temporales se descartan, de forma que las lecturas que se produzcan al terminar el FLUSH toman los datos de memoria. La directiva FLUSH es la única que permite que se pueda pasar un dato de un thread a otro thread. Para ello, en la hebra origen se hace una escritura a la variable compartida y luego un FLUSH a la variable. En la hebra destino debe hacerse un FLUSH de la variable y luego la lectura.

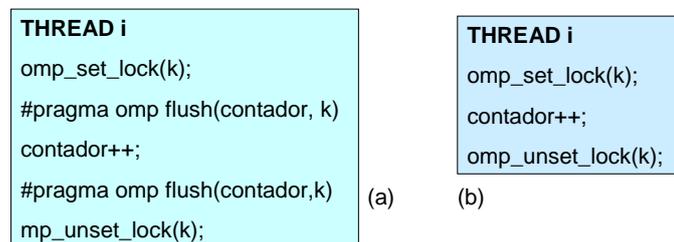


Figura 2. Implementación de un incremento de un contador en una sección crítica: (a) OpenMP 2.0, sin FLUSH implícito en el LOCK; (b) OpenMP 2.5, con FLUSH implícito.

La interacción de la arquitectura y la programación que tiene lugar debido al modelo de consistencia se pone de manifiesto al comparar la actualización de una variable mediante dos hebras sincronizadas a través de un LOCK en dos versiones diferentes de OpenMP. En la Figura 2.a se muestra la situación para la versión 2.0 donde es necesario incluir explícitamente FLUSH, mientras que en la versión 2.5, el FLUSH está implícito en el LOCK.

Según las estrategias de la Tabla 2, para motivar el aprendizaje de los modelos de consistencia de memoria es conveniente: (1) utilizar de ejemplos sencillos de accesos a memoria generados por procesos/hebras diferentes, que puedan ocasionar problemas según los distintos modelos de consistencia; y (2) el estudio de programas paralelos de memoria compartida con OpenMP que generen resultados diferentes si no se tiene en cuenta el modelo de consistencia de memoria que se usa. En [31] se hace referencia a las implicaciones del modelo de consistencia de memoria de OpenMP en los errores de programación, dado que cuando se lee una variable compartida sin anteponer un FLUSH, no está garantizado que el valor que se obtenga esté actualizado. Además, también habría que utilizar FLUSH en las hebras que han escrito en dicha variable con anterioridad. Afortunadamente, en muchos casos los errores no son aparentes en los programas porque muchas construcciones de OpenMP realizan FLUSHes implícitos. En muchas ocasiones, no se observan errores porque las variables se declaran como “volatile”, con lo que existe un FLUSH implícito delante de cada lectura y detrás de cada escritura de la variable en cuestión. Sin embargo, esto supone que el compilador no podrá aplicar muchas optimizaciones posibles a esas variables (por ejemplo, las relacionadas con la reordenación de accesos). Por eso, lo ideal es que el programador sea consciente del problema y utilice FLUSH (bien explícitamente a través de la correspondiente directiva o de una construcción OpenMP que la incluya implícitamente) solo cuando sea imprescindible.

Un aspecto a tener en cuenta respecto a la inclusión de los modelos de consistencia en la docencia de asignaturas de Estructura y Arquitectura de Computadores es que no se trata de un problema totalmente resuelto [28]. A los estudiantes se les puede enseñar alguna de las propuestas actuales, pero también se puede razonar en torno a sus limitaciones para poner de manifiesto que existen posibilidades de innovación dentro de este ámbito de conocimiento. De esta forma, también se incide positivamente en su motivación.

6 Conclusiones

Tras resumir algunas de las últimas conclusiones sobre el proceso de aprendizaje y después de revisar las características fundamentales de las teorías de la motivación más relevantes se ha puesto de manifiesto la importancia de la visión previa que los estudiantes tienen cuando inician el estudio de una asignatura y la diferencia entre motivación intrínseca y motivación extrínseca. De esta forma, las necesidades psicológicas de autonomía, competencia y socialización hacen que, si bien una fuerte

demanda de profesionales que prácticamente garantice un puesto de trabajo sea un factor importante de cara a motivar a los estudiantes (motivación extrínseca), también lo sean la relevancia de los problemas que se plantean en la disciplina en cuestión y su utilidad (motivación intrínseca).

Teniendo en cuenta esta distinción entre motivación intrínseca y extrínseca, las Secciones 3 y 4 se han dedicado a proporcionar argumentos que justifiquen el interés por la Ingeniería de Computadores. Dado que el análisis realizado en la Sección 3 ha puesto de manifiesto una cierta dificultad de proporcionar motivaciones extrínsecas intensas para los estudios de Ingeniería de Computadores frente a otras especialidades, en la Sección 4 se han analizado las condiciones de la evolución en las tecnologías de la información y las comunicaciones que pueden fundamentar la motivación intrínseca de los estudiantes hacia la Ingeniería de Computadores porque las competencias que ofrece permiten intervenir, de forma decisiva, en la implantación de nuevas soluciones a problemas esenciales de nuestra sociedad.

Para terminar el artículo, se han propuesto una serie de estrategias docentes útiles para impartir los contenidos de las asignaturas relacionadas con la Ingeniería de Computadores. Se trata de estrategias inspiradas en las teorías sobre la motivación resumidas en la Sección 2 cuya aplicación han ilustrado utilizando, como ejemplos, la docencia del lenguaje ensamblador y la consistencia de memoria. Se hace hincapié en que los estudiantes deben contemplar los contenidos de la asignatura como relevantes para la profesión, y como asequibles, sin tener por ello que renunciar a impartir conceptos o problemas complejos que, por otra parte, son los que hacen que el estudiante llegue a ser consciente de la utilidad de la asignatura para su futuro profesional.

Referencias

1. Tanenbaum, A.S.: "Sistemas Operativos Modernos" (3ª Edición). Pearson Prentice Hall. México 2009. (ISBN 978-607-442-046-3).
2. Ambrose, S.A.; Bridges, M.W.; Di Pietro, M.; Lovett, M.C.; Norman, M.K.: "How learning works: Seven research-based principles for smart teaching". Jossey-Bass, San Francisco, 2010.
3. Bransford, J.D.; Brown, A.L.; Cocking, R.R. (Eds.): "How People Learn: Brain, Mind, Experience, and School". Nat. Academy Press, 2001.
4. Torres-Ayala, A. T.; Herman, G.L.: "Motivating learners: A primer for Engineering Teaching Assistants". American Society of Engineering Education (ASEE) Annual Conference, San Antonio, TX, June 10-13, 2012.
5. Pintrich, P. R.: "Motivation and Classroom learning". Handbook of psychology, pp.103-122, Wiley Online Library, 2003.
6. Oudeyer, P.Y.; Kaplan, F.: "What is intrinsic motivation?. A typology of computational approaches". Frontiers in Neurobotics, Vol.1, pp. 1-14. Noviembre, 2007.
7. Pink, D.H.: "La sorprendente verdad sobre qué nos motiva". Gestión'2000, 2010.
8. ACM/AIS/IEEE-CS (The Joint Task Force for Computing Curricula 2005): "Computing Curricula 2005. The Overview Report". Septiembre, 2005. (<http://www.acm.org/education/curricula-recommendations>)
9. Grado en Ingeniería Informática de la Universidad de Granada: <http://grados.ugr.es/informatica/pages/infoacademica>
10. SANDETEL, Análisis del Sector TIC Andaluz en 2010: http://www.sandetel.es/index.php?option=com_content&task=view&id=39&Itemid=65.

11. SANDETEL, Análisis de mercado para áreas de actividad relacionadas con las TIC: http://www.sandetel.es/index.php?option=com_content&task=view&id=42&Itemid=68.
12. Asociación de Empresas de Electrónica, Tecnología de la Información y Telecomunicaciones de España (AETIC): "Informe anual del sector español de Electrónica, Tecnologías de la Información y Telecomunicaciones", 2010.
13. Buyya, R.; Yeo, C.; Venugopal, S.: "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as Computing Utilities". Proc. of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC'08), pp.5-13, 2008.
14. SANDETEL, Interoperabilidad de las tecnologías de telecomunicación en escenarios de convergencia: http://www.sandetel.es/index.php?option=com_content&task=view&id=162
15. AMETIC, Accenture: "Communications, Media & Technology. Un estudio publicado por el Centro de Alto Rendimiento de Accenture (CAR)", 2012.
16. HIPEAC Roadmap 2011/2012: <http://www.hipeac.net/system/files/hipeac-roadmap2011.pdf>
17. McGettrick, A.; Theys, M.D.; Soldan, D.L.; Srimani, P.K.: "Computer engineering curriculum in the new millennium". IEEE Transactions on Education, Vol. 46, No.4, pp.456-462. Noviembre, 2003.
18. CISCO Visual Networking Index (VNI): www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html
19. Cooney, M.: "Gartner: Top 10 strategic technology trends for 2013". Octubre, 2012: <http://www.computerworlduk.com/news/applications/3406773/gartner-top-10-strategic-technology-trends-for-2013/>
20. Mycroft A.: "Programming Languages Design and Analysis motivated by Hardware Evolution". H. Riis and G Filé (Eds.), SAS 2007, LNCS 4643, pp.18-33, 2007.
21. Bernstein, D.: "Software enablement for multicore architectures". [http://ce.et.tudelft.nl/~carlo/File_index/Symposium/02_Bernstein_\[Vassiliadis_Symposium_2007\].pdf](http://ce.et.tudelft.nl/~carlo/File_index/Symposium/02_Bernstein_[Vassiliadis_Symposium_2007].pdf), 2007.
22. Patt, Y.: "Ten Commandments for Good Teaching". http://users.ece.utexas.edu/~patt/Ten_commandments/, 2013.
23. Litzinger, T.A.; Lattuca, L. R.; Hadgraft, R.G.; Newstetter, W.C.: "Engineering education and the development of expertise". Journal of Engineering Education, Vol. 100, No.1, pp.123-150. Enero, 2011.
24. McConnell, J.J.: "Active and cooperative learning: tips and tricks (part I). SIGCSE Bulletin, 37, 2, pp. 27-30, 2005.
25. Gerber, R.: "The Software Optimization Cookbook. High Performance Recipes for the Intel Architecture". Intel Press, 2002.
26. Barlett, J.: "Programming from the ground up". www.barlettpublishing.com, 2003.
27. <http://joelonsoftware.com/>
28. Adve, S. V.; Boehm, H.-J.: "Memory models: a case for rethinking parallel languages and hardware". Communications of the ACM, Vol. 53, no.8, pp.90-101. Agosto, 2010.
29. OpenMP Architecture Review: "OpenMP Application Programming Interface" (Version 3.0), www.openmp.org, 2008.
30. Hoeflinger, J.P.; Supinski, B.R.: "The OpenMP Memory Model". Proceedings of the First International Workshop on OpenMP (IWOMP 2005), 2005.
31. Süß, M.; Leopold, C.: "Common Mistakes in OpenMP and how to avoid them. A collection of best practices". OpenMP Shared Memory Parallel Programming, LNCS vol. 4315, pp. 312-323, 2008.

Aprendizaje Cooperativo y Portafolio Digital de Grupo: Desarrollo de Competencias

C. Gil¹, R. Baños², M.G. Montoya¹, R.I. Herrada³, F.G. Montoya⁴

¹Dpto. de Informática, ceiA3, Universidad de Almería. Carretera de Sacramento s/n, E-04120 Almería (España), {cgil,dgil}@ual.es

²Dpto. de Arquitectura y Tecnología de Computadores, CITIC-UGR, Universidad de Granada. C/Periodista Daniel Saucedo s/n, E-18071 Granada (España), rbanos@ugr.es

³Dpto. de Didáctica y Organización Escolar, Universidad de Murcia. Campus de Espinardo, E-30100 Murcia (España), rherrada@um.es

⁴Dpto. de Ingeniería. Universidad de Almería. Carretera de Sacramento s/n, E-04120 Almería (España), pagilm@ual.es

Resumen. Con el objetivo de determinar la efectividad en el uso del Aprendizaje Cooperativo y el uso del Portafolio Digital de grupo en el proceso de enseñanza-aprendizaje-evaluación, se han analizado los resultados obtenidos en relación al número de alumnos que han superado las diferentes asignaturas en los últimos años de aplicación de estas metodologías. Los resultados muestran que el uso de dichas metodologías conlleva una mayor motivación del estudiante hacia la materia, y un descenso del número de abandonos. Igualmente la motivación del profesorado hacia la enseñanza se ha visto reforzada por la mejora del rendimiento obtenido por la mayoría de los grupos de aprendizaje cooperativo que han desarrollado y completado su portafolio.

Palabras Clave: Aprendizaje Cooperativo, Competencias, Portafolio Digital, Arquitectura de Computadores, TIC.

Abstract. With the aim of determining the effectiveness of Cooperative Learning in combination with the Group Digital Portfolio in the teaching-learning-assessment process, we have analyzed the results in relation to the number of students who approved the subject in the years in which of applying these methodologies have been applied. The results show that the use of these methodologies involves greater student motivation toward the subject, and a decrease in the number of dropouts. Moreover, lecturers' motivation toward learning has been reinforced by the improvement of performance as most cooperative learning groups who have developed their portfolio.

Keywords: Cooperative Learning, Competencies, Digital Portfolio, Computer Architecture, ICT.

1 Introducción

En el contexto actual, el trabajo se organiza cada vez más en equipos multidisciplinares y multiculturales capaces de trabajar en contextos caracterizados por los constantes cambios a los que están sometidos. En estos equipos se valora especialmente la participación, la comunicación, y el trabajo en común de tipo cooperativo. Por ello, resulta necesario que alumnado y profesorado desarrollen la cooperación en las aulas como medio para prepararse para la sociedad actual y del futuro. La primera consecuencia práctica que surge es la necesidad real de adecuar el currículo formativo a las claves de una educación basada en competencias. Sin embargo, el factor fundamental para poder desarrollar dichas competencias consiste en utilizar las metodologías docentes más adecuadas para cada una de ellas, y en cualquier caso deberán ser metodologías activas que impliquen una mayor participación de los estudiantes.

De estas metodologías activas, el *Aprendizaje Cooperativo (AC)* es uno de los procedimientos de enseñanza-aprendizaje más utilizados por aquellos docentes que desean maximizar el aprendizaje de los estudiantes, y que dicho aprendizaje sea retenido a largo plazo [1], incluso en aquellas materias altamente complejas y difíciles de entender y dominar.

La cooperación consiste en trabajar juntos para alcanzar objetivos comunes. En una situación cooperativa, los individuos procuran obtener resultados que sean beneficiosos para ellos mismos y para todos los demás miembros del grupo. El aprendizaje cooperativo es el empleo didáctico de grupos reducidos en los que los alumnos trabajan juntos para maximizar su propio aprendizaje y el de los demás. Este método contrasta con el aprendizaje individualista y competitivo, en el que los estudiantes trabajan por su cuenta para lograr metas de aprendizaje desvinculadas de las de los demás alumnos [2]. Por ello, es muy importante que el trabajo cooperativo cuente con técnicas de evaluación apropiadas, ya que la evaluación es un modo excelente de hacer que los estudiantes empleen tiempo en sus tareas. Sin embargo, una de las principales dificultades para mejorar la motivación por aprender radica en los modos de evaluación tradicionalmente utilizados por los profesores, normalmente exámenes centrados en la identificación de resultados producidos por el alumno de forma puntual con ocasión de la evaluación [3]. El efecto de estas prácticas es, por lo general, que el alumno estudia para aprobar y no para aprender, actitud que dificulta la comprensión y adquisición de competencias así como el desarrollo de una actitud reflexiva que potencie la capacidad de autorregulación y aprendizaje autónomo. Así, la evaluación constituye uno de los puntos débiles de la docencia universitaria [4].

Pese a ello, existen soluciones que permiten incorporar la autorreflexión, lo que refuerza el aprendizaje del alumno debido a que le proporciona oportunidades para autoevaluar su propio crecimiento. En este sentido, el *Portafolio* es un método de enseñanza, aprendizaje y evaluación que consiste en la aportación de producciones de diferente índole por parte del estudiante a través de las cuáles se pueden juzgar sus capacidades en el marco de una disciplina o materia de estudio. Estas producciones informan del proceso personal seguido por el estudiante, permitiéndole a él y a los demás ver sus esfuerzos y logros, en relación a los objetivos de aprendizaje y criterios de evaluación establecidos previamente [5]. Más concretamente, el auge de Internet

ha permitido la extensión en el uso del *Portafolio digital* como método de enseñanza y aprendizaje, que, gracias a su naturaleza gráfica y habilidad para soportar enlaces entre distintas evidencias digitalizadas, proporciona al alumnado la posibilidad de integrar los aprendizajes de un modo positivo, progresivo y consciente con un gran potencial atractivo [5].

La aplicación práctica del trabajo cooperativo en conjunción con el *Portafolio digital* de grupo, se ha llevado a cabo en varias asignaturas de Ingeniería Informática así como en el master oficial de Técnicas Informáticas Avanzadas y en el Master de Profesorado de Secundaria, impartidos en la Universidad de Almería. La razón fundamental del cambio metodológico que venimos desarrollando en los últimos años tiene como objetivo final implicar a los estudiantes de forma activa en su proceso de aprendizaje, además de desarrollar las competencias específicas y transversales de cara a su formación integral.

2 Metodologías de Trabajo en Equipo

El trabajo en equipo no es sólo un recurso metodológico para enseñar y aprender los contenidos de las distintas áreas, sino también algo que los alumnos deben aprender, como un contenido más, y que, por lo tanto, debe enseñarse de una forma tan sistematizada, al menos, como se enseñan los demás contenidos. Por ello, podríamos suscribir la afirmación que, sobre este aspecto, hacen Johnson y Johnson [6]:

La capacidad de todos los alumnos de aprender a trabajar cooperativamente con los demás es la piedra clave para construir y mantener matrimonios, familias, carreras y amistades estables. Ser capaz de realizar habilidades técnicas como leer, hablar, escuchar, escribir, calcular y resolver problemas es algo valioso pero poco útil si la persona no puede aplicar estas habilidades en una interacción cooperativa con las otras personas en el trabajo, en la familia y en los entornos comunitarios. La manera más lógica de enfatizar el uso del conocimiento y las habilidades de los alumnos dentro de un marco cooperativo, tal como deberán hacer cuando sean miembros adultos de la sociedad, es dedicar mucho tiempo al aprendizaje de estas habilidades en relaciones cooperativas con los demás.

No hay nada mágico en el trabajo en equipo. Algunos tipos de grupos facilitan el aprendizaje de los alumnos y mejoran la calidad de vida en el aula. Otros entorpecen el aprendizaje y provocan insatisfacción y falta de armonía en la clase. Para manejar con eficacia los grupos de aprendizaje, el docente debe saber qué es y qué no es un grupo cooperativo.

2.1 Aprendizaje Cooperativo

El objetivo de poner en marcha metodologías activas como *Aprendizaje Cooperativo* ha sido lograr la implicación y participación de todos los alumnos en su propio proceso de aprendizaje a través de objetivos comunes. Se ha desarrollado una estrategia de enseñanza centrada en el alumno que permite

producir conocimiento desde la interacción en un grupo de aprendizaje cooperativo. También se pretende estimular y desarrollar competencias como el trabajo en equipo, la comunicación oral y la evaluación.

Cooperar significa trabajar juntos para lograr objetivos compartidos. En las actividades cooperativas, los individuos buscan resultados que resulten beneficiosos para sí mismos y, al mismo tiempo, para todos los otros integrantes del grupo. El *Aprendizaje Cooperativo* consiste en el uso educativo de pequeños grupos que permiten a los estudiantes trabajar juntos para mejorar su propio aprendizaje y el de los demás. Para que los grupos de aprendizaje cooperativo sean realmente eficaces y se acerquen en lo posible a los niveles de mayor rendimiento, deben incorporarse a cada clase cinco elementos básicos de dicho sistema de enseñanza-aprendizaje: la interdependencia positiva, la responsabilidad individual, la interacción cara a cara, la enseñanza de prácticas interpersonales, y la autoevaluación grupal [7].

El aprendizaje cooperativo comprende diferentes grupos de aprendizaje. Por un lado, encontramos a los grupos informales, que se desarrollan en el periodo de una clase. Por otro lado, los grupos formales y los grupos base, que se tienen una mayor duración en el tiempo, en nuestro caso los usaremos indistintamente, y comprenden un cuatrimestre o un curso. Nos centraremos en estos últimos que son los que se han analizado en las asignaturas tratadas. En los grupos formales de aprendizaje cooperativo los estudiantes trabajan juntos para lograr objetivos comunes, asegurándose de que ellos mismos y sus compañeros de grupo completen la tarea de aprendizaje asignada. Cualquier tarea, de cualquier materia y dentro de cualquier programa de estudios, puede organizarse en forma cooperativa. Cualquier requisito del curso puede ser reformulado para adecuarlo al aprendizaje cooperativo formal [2].

Cuando se emplean grupos formales de aprendizaje cooperativo, el docente debe: (a) especificar los objetivos de la clase, (b) tomar una serie de decisiones previas a la enseñanza, (c) explicar la tarea y la interdependencia positiva a los alumnos, (d) supervisar el aprendizaje de los alumnos e intervenir en los grupos para brindar apoyo en la tarea o para mejorar el desempeño interpersonal y grupal de los alumnos, y (e) evaluar el aprendizaje de los estudiantes y ayudarlos a determinar el nivel de eficacia con que funcionó su grupo. Los grupos formales de aprendizaje cooperativo garantizan la participación activa de los alumnos para trabajar el material, explicarlo, resumirlo e integrarlo a las estructuras conceptuales existentes [2].

Es importante que los grupos se formen desde la diversidad de talentos y posibilidades intelectuales [8]. Esto enriquecerá el intercambio y preparará el terreno para el salto cognitivo de los más lentos.

2.2 Portafolio Digital de Grupo

El *Portafolio* puede usarse para el desarrollo y valoración del conocimiento de una asignatura, para la adquisición de habilidades de enseñanza y prácticas reflexivas, así como para la preparación profesional y vocacional. En nuestro caso, el uso del *Portafolio* digital de grupo en un entorno de aprendizaje cooperativo ha permitido trabajar competencias transversales que difícilmente podríamos desarrollar a través de la clase expositiva tradicional y la posterior evaluación a través del examen final.

Un *Portafolio* es una colección de trabajos que incluyen los logros individuales o grupales, tales como resultados de tareas genuinas, la evaluación del proceso, test convencionales o muestras de trabajo. Es decir, documenta los logros alcanzados a lo largo del tiempo. Generalmente el individuo elige el tipo de trabajo que le sirva mejor para expresar su éxito, así como para demostrar su aprendizaje respecto a un objetivo particular como podría ser la certificación o la evaluación tanto sumativa como formativa. De forma genérica, las principales características del *Portafolio* son las siguientes:

- Es una herramienta reflexiva que permite almacenar y preservar evidencia del crecimiento del alumno, profesor, institución, proyecto, etc.
- Es una colección de trabajos que nos permite conocer lo que el individuo, equipo de trabajo o institución sabe y puede hacer.
- Su contenido es auténtico, con un objetivo representativo y auto-dirigido.
- Constituyen una alternativa para evaluar, certificar, informar, promover, etc.

El *Portafolio* no es un fin en sí mismo, sino que más bien gracias a él se consigue un aprendizaje continuo y exitoso, el cual se debe a la asociación de la evaluación con las prácticas y procesos de enseñanza llevados a cabo con dicho método [3].

El desarrollo del *Portafolio* implica documentar, no sólo los logros conseguidos, sino también las autoevaluaciones, las estrategias aplicadas, y el análisis sobre las experiencias de aprendizaje, por lo que es más que una simple colección de tareas.

En la actualidad, las TIC aportan muchas potencialidades a la hora de diseñar y elaborar un *Portafolio*. En esta línea, las TIC pueden ayudar en la elaboración de las evidencias que conforman el *Portafolio* y también pueden actuar como plataforma base del proceso de enseñanza y aprendizaje. Una de las principales ventajas del *Portafolio* digital es que toda la información puede estar disponible para todos los estudiantes y pueden aprender a desarrollar las tareas y actividades aprendiendo con otros compañeros, pueden tutorizar a otros grupos cuando les plantean cuestiones sobre sus actividades, además de poder participar en el proceso de co-evaluación.

3 Experiencia Práctica en Arquitectura de Computadores

Como se ha comentado anteriormente, el contexto donde han tenido lugar las experiencias ha sido en las asignaturas de 4º curso de Ingeniería Informática: Arquitecturas Especializadas y Arquitectura de Computadores, cuya impartición finaliza este año y en las asignaturas de master oficiales como “Optimización Multiobjetivo: Estrategias de Paralelismo” del master oficial en Técnicas Informáticas Avanzadas y “Metodologías Activas de Trabajo en Equipo” del master oficial de Profesorado de Educación Secundaria de la Universidad de Almería.

Aunque las experiencias han sido muy similares, nos centraremos en la asignatura de “Arquitectura de Computadores” de 4º curso de Ingeniería Informática. Hay que resaltar que la experiencia realizada no ha estado dentro de ningún plan piloto y que, por tanto, se puede considerar como cambios metodológicos en asignaturas de un plan de estudios que se encuentra en proceso de extinción. Esto demuestra que no es

necesario cambiar un plan de estudios para introducir cambios en la motivación y mejora de resultados de cualquier asignatura.

En algunas de las clases de teoría se ha mezclado la clase expositiva con la resolución de problemas a través de grupos informales de aprendizaje cooperativo creados en cada clase. El mero hecho de hacer un cambio entre una explicación y la realización de un problema hace que los alumnos retomen el control. Discuten por parejas y en algunos casos toda la clase debate sobre posibles soluciones.

En la mayor parte de las sesiones se ha trabajado con grupos formales de aprendizaje cooperativo. Es en estos grupos formales de aprendizaje cooperativo donde centramos nuestra experiencia.

3.1 Aprender a Trabajar en Equipo en Arquitectura de Computadores

No aplicar un método que no nos acaba de satisfacer, evidentemente, es muy legítimo. Pero si el trabajo en equipo, además de un método, es también un contenido que deben aprender los alumnos, no podemos eludir su enseñanza diciendo que los alumnos no saben trabajar en equipo, sino que tendremos que enseñárselo, identificando qué es lo que no funciona, cambiando lo que sea necesario, e insistiendo tanto como haga falta hasta que lo aprendan [9]. Este sería el caso además, cuando la competencia transversal “trabajo en equipo” forma parte de las competencias asignadas a una asignatura. En estos años hemos detectado la importancia y la necesidad de enseñar a nuestros alumnos a cooperar y a trabajar en equipo como técnica clave para que el Aprendizaje Cooperativo sea mucho más eficaz y productivo en el aprendizaje y crecimiento integral de nuestros estudiantes. Para ello hemos elaborado un plan de trabajo para enseñar a cooperar y trabajar en equipo, como necesidad previa para el mejor funcionamiento de los grupos cooperativos. Este plan, adaptado de [9,10], incluye distintas actividades y dinámicas para cubrir los siguientes objetivos:

- Mostrar a los alumnos la importancia de saber trabajar en equipo.
- Despertar el interés de los alumnos para trabajar en equipo.
- Enseñar a los alumnos en qué consiste el trabajo en equipos cooperativos y como se puede mejorar.
- Ayudar a los alumnos a organizar su equipo.
- Desarrollar habilidades para la resolución de conflictos.
- Evaluar las actividades de trabajo en equipo.

Una vez que aprendidas y practicadas estas habilidades en una determinada materia, ya estarían en disposición de trabajar en equipo o cooperativamente en cualquier otra asignatura.

3.2 Trabajar en Equipo para Aprender Arquitectura de Computadores

Una vez que a los alumnos se les entrena en las habilidades y actitudes que hacen que el trabajo en equipo funcione y que les permitan diferenciar el trabajo en grupo

tradicional del trabajo en equipo, se trata de desarrollar las actividades de aprendizaje de las competencias genéricas y específicas de la materia utilizando grupos cooperativos [11].

Para aprender las competencias específicas de la materia se llevan a cabo un conjunto de actividades, cada una de las cuales corresponde a un bloque temático de la asignatura. Para cada actividad hay una parte del trabajo que se realiza de forma individual y otra parte que se realiza en equipo. Realizar una parte del trabajo de forma individual desarrolla la responsabilidad, ingrediente fundamental del aprendizaje cooperativo. Para promover la responsabilidad individual y la interdependencia positiva es necesario que, en las actividades que diseñe el profesor, los alumnos dependan de sus compañeros de equipo, ya sea para ayudarlos o para recibir ayuda. Además, al final de la actividad, se hará una pregunta al azar a cualquiera de los miembros del equipo para asegurar la responsabilidad individual.

Durante las sesiones de teoría y de laboratorio, se ocupa una parte del tiempo para que los grupos realicen parte de la actividad propuesta en horario de clase, incluyendo trabajo individual y en grupo mediante la técnica del puzzle [2], pero además deben dedicar algún tiempo más a trabajar fuera del aula en las diferentes actividades. En concreto, cada actividad ha tenido una duración aproximada de tres semanas. El plan de trabajo para cada actividad, que coincidirá con los temas de la asignatura, se estructura de la siguiente forma:

- Previo a la primera sesión presencial de realización de la actividad, a cada alumno de un grupo se le asigna una actividad diferente, es decir uno realiza un mapa conceptual, otro realiza un glosario de términos, y el otro alumno (en el caso de grupos de tres) preguntas tipo test, y de esta se consigue que lean y trabajen el tema antes de la resolución de problemas.
- En la primera sesión presencial de cada actividad se entrega el resto del trabajo a cada estudiante del grupo cooperativo. Cada actividad está estructurada de forma que hay una parte de trabajo individual diferente para cada miembro del grupo (una cuestión tipo test y un problema diferente para cada alumno del grupo) y una parte común para todo el grupo. Para ello, en dicha sesión presencial, primero individualmente dedican una parte del tiempo a resolver el problema y la cuestión. Pasado ese tiempo se reúne el grupo de expertos (puzzle, alumnos de distintos grupos que tienen asignado el mismo problema) para comparar soluciones y aclarar dudas. Posteriormente, regresan al grupo base original para que cada miembro explique su parte del trabajo al resto de compañeros. La parte grupal de la actividad se realiza fuera de clase y tienen que colaborar todos los miembros del grupo base, teniendo en cuenta los conocimientos adquiridos en la primera parte por cada uno de ellos. Todos tienen que conocer y dominar el trabajo de todo el grupo, ya que pueden ser elegidos aleatoriamente para exponer cualquier parte del trabajo. Finalmente, todo el material se pone antes de la próxima sesión en el *Portafolio* del grupo (versión sin corregir).
- En la segunda sesión presencial se realiza co-evaluación cíclica (entre los diferentes grupos) del trabajo realizado. La tarea de evaluar a otros grupos forma parte de las competencias transversales de la materia y, por tanto, también será evaluada. Previamente se facilitarán rúbricas para poder realizar correctamente la tarea de evaluación. Una vez realizado dicho trabajo, cada grupo le explica al grupo al que

ha evaluado qué aspectos se han realizado correctamente y cuáles deben mejorarse y corregirse. La versión corregida del trabajo debe colocarse en el *Portafolio* en los plazos establecidos, siempre antes de la siguiente sesión.

- En la tercera sesión presencial de la actividad se realiza una presentación oral o una webquest de un trabajo asignado al grupo. Todos los miembros del grupo deben exponer parte del trabajo.
- Por último, cada dos actividades, se realiza una prueba individual de mínimos a todos los alumnos.

Después de la prueba individual dará comienzo una nueva actividad y el ciclo de tres semanas se vuelve a repetir para cada actividad propuesta, intentando además que en la fase de interacción entre los diferentes grupos (puzzle), los alumnos que interaccionan sean distintos.

Es necesario superar los cuatros temas que se evalúan en dos pruebas de mínimos que se realizarán a lo largo del curso. Hay una segunda oportunidad para recuperar cada tema en el examen final.

3.3 Estructura del Portafolio Digital de Grupo

El uso del *Portafolio* digital de grupo ha sido una herramienta más en el proceso de enseñanza-aprendizaje a través de grupos cooperativos. En los últimos años, el *Portafolio* se ha diseñado haciendo uso de la plataforma digital de apoyo a la docencia WebCT [12]. El último curso (2011-2012) también se ha utilizado la plataforma Moodle, dado que para la misma hemos desarrollado un módulo de *Portafolio*.

En la plataforma WebCT, a los alumnos se les facilita un archivo *html* con la estructura básica y los apartados que deben contener todos los *Portafolios* así como una zona para las fotos y nombres de los integrantes de cada grupo.

El *Portafolio* se ha organizado en diferentes apartados que han ido dando cuenta de todo el trabajo de aprendizaje y reflexión llevado a cabo por cada grupo. Los apartados definidos para todos los *Portafolios* han sido los siguientes:

- **Objetivos:** En este apartado se presentan al estudiante los objetivos que se pretenden alcanzar a través de la realización de las diferentes actividades, así como la importancia del *Portafolio* para el seguimiento de la asignatura, reflexión del propio trabajo y autoevaluación del grupo.
- **Problemas:** En este apartado cada grupo debe colocar las diferentes versiones (sin corregir y corregida) de los problemas asignados en cada actividad. Se ofrece una rúbrica sobre los criterios a valorar para la correcta resolución de problemas.
- **Glosario.** En este apartado se colocan los términos más importantes asociados con cada tema de la asignatura y que será asignado a un alumno diferente del grupo en cada actividad.
- **Mapas Conceptuales.** En este apartado se coloca el mapa conceptual que es realizado en cada tema por un alumno diferente del grupo. Suelen utilizar herramientas tipo *CmapTools* para realizarlo e incluirlo en la plataforma.

- **Preguntas test.** En este apartado se incluyen las preguntas tipo test a realizar por un alumno diferente del grupo en cada actividad. Suelen utilizar herramientas como el software *Hot Potatoes*, que permite poder responder a las cuestiones tipo test como si de una prueba o examen se tratara a través de una página web.
- **Trabajos.** En este apartado los estudiantes colocan los trabajos asignados sobre cada tema, así como cualquier aportación propia que consideren importante en el aprendizaje de la materia.
- **Competencias:** Este apartado contiene las rúbricas sobre las diferentes competencias a evaluar: trabajo en equipo y comunicación oral. Los estudiantes colocan en primer lugar el reglamento del grupo, que realizan en los primeros días de creación de los grupos cooperativos y que contiene las normas de funcionamiento del grupo así como las autoevaluaciones del mismo.

A modo ilustrativo de lo anteriormente expuesto, en la Figura 1 se muestra la estructura del *Portafolio* digital que el profesorado le suministra a todos los estudiantes, indicándoles con ello, las secciones que deben tener todos los *Portafolios* en la asignatura de Arquitectura de Computadores.



Figura 1. Imagen de la estructura general que todos los *Portafolios*.

Las Figuras 2, 3 y 4 muestran un ejemplo del resultado final del *Portafolio* digital de uno de los grupos cooperativos para distintas partes del mismo. En la parte izquierda de la Figura 2 se muestra la estructura del *Portafolio* que previamente les facilitó el profesorado y que se muestra en la Figura 1 (Objetivos, Problemas, Glosarios, Trabajos, etc). La Figura 3 muestra la parte del *Portafolio* correspondiente a los problemas que han venido trabajando a lo largo de las actividades durante el curso. Se muestra tanto la versión sin corregir como la versión corregida. La Figura 4 muestra la parte del *Portafolio* correspondiente a los trabajos, y en concreto uno de ellos que consistía en hacer una Webquest sobre topología de redes estáticas.

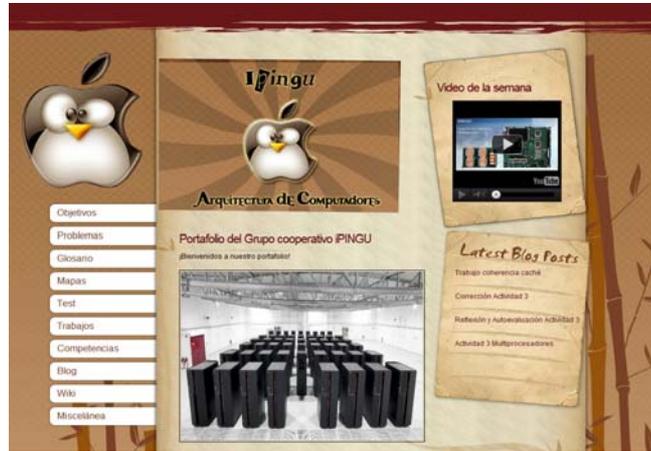


Figura 2. Imagen de la PAGINA INICIAL del Portafolio del grupo Ipingu.

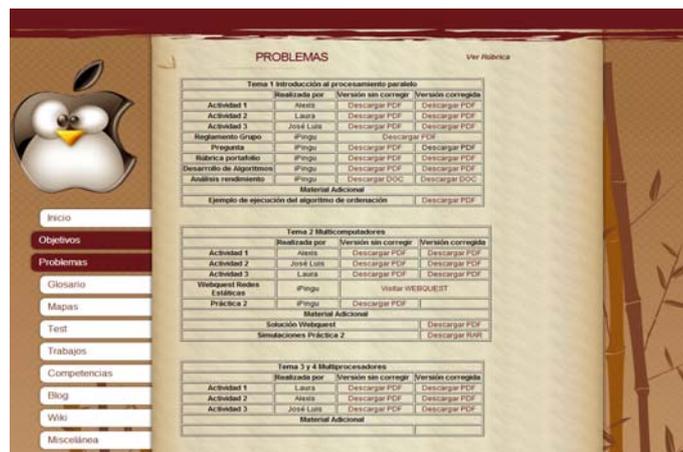


Figura 3. Imagen de la sección PROBLEMAS del Portafolio del grupo Ipingu.

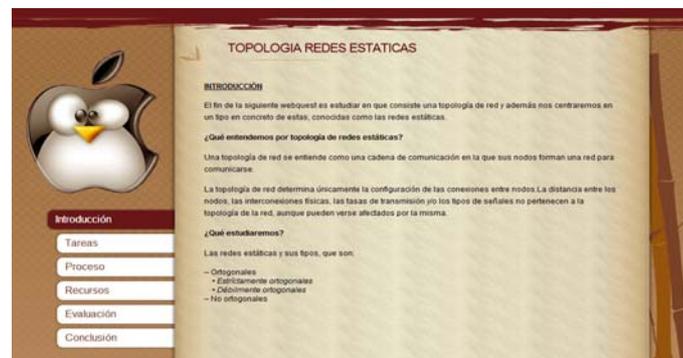


Figura 4. Imagen de la sección de TRABAJOS del Portafolio del grupo Ipingu.

4 Evaluación

Se ha optado por establecer un método de evaluación continua, lo que implica necesariamente un seguimiento y control del avance de cada estudiante a lo largo de la asignatura. Se puntúa de la siguiente forma:

La nota de grupo incluye:

- Las actividades en grupo ponderan un 30% (glosarios, cuestiones, problemas, prácticas, trabajos, etc.). Se valora un 15% sólo por entregarlas en la fecha acordada aunque no estén correctas y un 15% por entregarla corregida en la fecha establecida. Si no se realizan todas las actividades no se superará la asignatura.
- Las preguntas al azar a cualquier miembro del grupo sobre el trabajo presencial realizado en las actividades pondera un 10% y la nota será para todo el grupo.
- El diseño y desarrollo del *Portafolio* digital de grupo sobre el que se colocarán todas las actividades tendrá una ponderación del 10% en la nota.
- La adquisición de competencias suponen un 10% de la nota, y comprende el trabajo en equipo (5%), evaluado mediante la asistencia regular al trabajo del grupo en clase, autoevaluaciones de compañeros, observación de la profesora, etc. Y la comunicación oral (5%), evaluada mediante la presentación de trabajos.

La nota individual incluye:

- Las pruebas individuales ponderarán un 40% y se valorarán aspectos de conocimientos mínimos de la asignatura. Para superar la asignatura es necesario superar 7 de los 8 conocimientos mínimos evaluados en dos pruebas. La calificación de la componente de conocimientos mínimos variará una vez superados los 7 ó 8 mínimos dependiendo del número de convocatorias utilizadas [7].

5 Resultados de la experiencia

Debido a la importancia que para nosotros ha supuesto el uso del Portafolio Digital combinado con los grupos de aprendizaje cooperativo y de cara a una utilización más estándar del mismo, un resultado significativo de esta experiencia, ha sido nuestro interés por desarrollar un módulo *Portafolio* genérico que se pueda usar en plataformas de libre distribución. Para ello, durante el pasado curso académico 2011-2012, se ha diseñado y puesto en marcha una nueva herramienta para incluir el *Portafolio* digital en la plataforma Moodle. Esta plataforma es de libre distribución y hemos empezado a usarla en las asignaturas de Master. Puesto que Moodle permite la creación de grupos de trabajo, lo que se ha hecho es diseñar un módulo que permite al profesor crear el *Portafolio* de todos los grupos de forma automática a partir de la creación de dichos grupos. Este *Portafolio* incluye la foto y nombre de los alumnos pertenecientes al grupo, un foro accesible únicamente para los miembros del grupo, secciones para organizar el trabajo de los estudiantes, y un área para manejar archivos. A diferencia del *Portafolio* que hemos utilizado en WebCT, donde son los alumnos los que realizan todo el diseño del *portafolio* en ficheros HTML (Figuras 2, 3

y 4), a través de este nuevo módulo, hemos conseguido descargar a los alumnos de esa tarea, ya que ahora, en Moodle es el sistema el que lo genera automáticamente, con el consiguiente ahorro de tiempo para los estudiantes (Figuras 5, 6, y 7). El diseño de todos los *Portafolios* en Moodle es el mismo y sólo varían las secciones personales que pueden incluir los estudiantes.

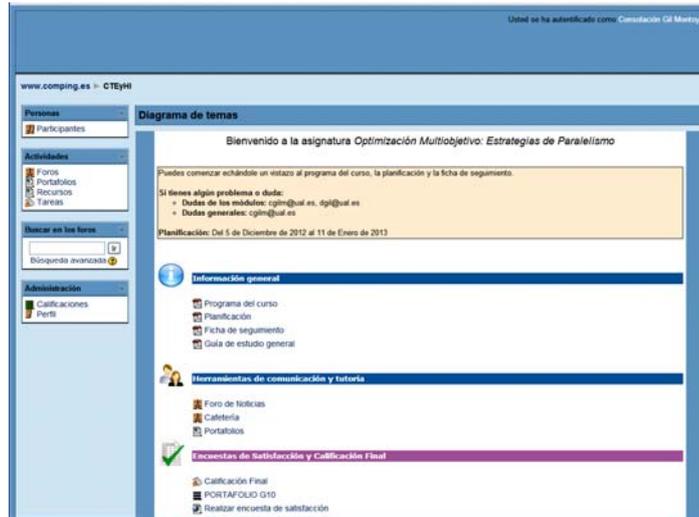


Figura 5. Plataforma Moodle para la asignatura OMEP del Master TIA.

La Figura 5 muestra la página inicial de la asignatura Optimización Multiobjetivo: Estrategias de Paralelismo (OMEP) del Master Técnicas Informáticas Avanzadas (TIA). En la parte izquierda de dicha figura se observa el acceso al *Portafolio*.



Figura 6. *Portafolio* del grupo cooperativo de las profesoras.

La Figura 6 muestra la imagen del *Portafolio*. En este caso se ha optado por poner el *Portafolio* de las profesoras, para evitar poner la foto de los estudiantes. En la parte izquierda aparecen las secciones que las profesoras han diseñado para todos los grupos, además del manejo de archivos. En la parte derecha se encuentran las secciones de los alumnos y el acceso al foro específico del grupo.

Por último, la Figura 7 muestra el acceso a una de las secciones de trabajo del grupo, en este caso, la parte que muestra el reglamento del grupo.

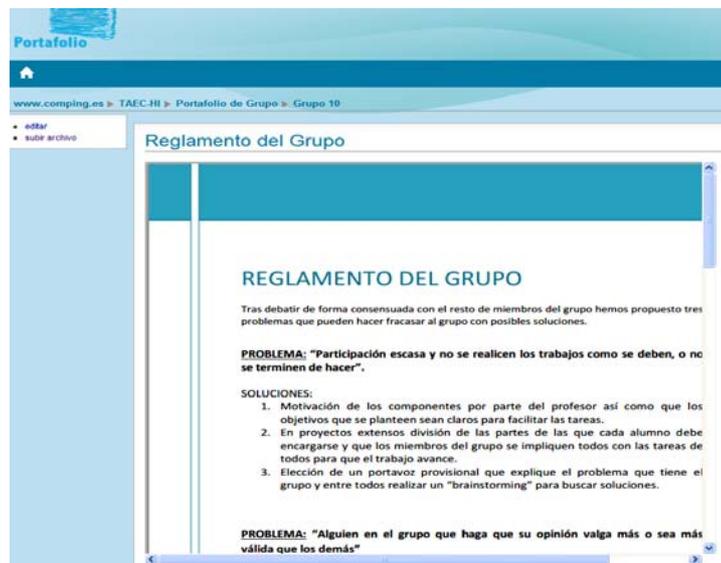


Figura 7. Imagen de la sección de TRABAJOS del *Portafolio* digital de un grupo.

Los resultados obtenidos a lo largo del tiempo denotan claramente que el número de alumnos que de forma voluntaria optan por esta metodología, en detrimento de jugárselo todo en el examen final, se ha ido incrementando progresivamente. Esto denota que los estudiantes están más motivados por la asignatura, discuten, hablan, preguntan y algo fundamental para el profesorado, es que siguen la asignatura al día, sin acumular la materia para estudiar en los días previos al examen final.

Los datos concretos, presentados en la Figura 8 muestran como, por un lado, el porcentaje de estudiantes que, en lugar de desistir en preparar la asignatura, la preparan y se presentan a los exámenes aumenta considerablemente desde el momento en el que se empiezan a aplicar estas metodologías activas. De igual forma, el número de aprobados también ha sido bastante sorprendente, tal y como muestra la Figura 8, ya que se ha pasado de un 60% de aprobados en el sistema tradicional a un porcentaje que ronda el 90%.

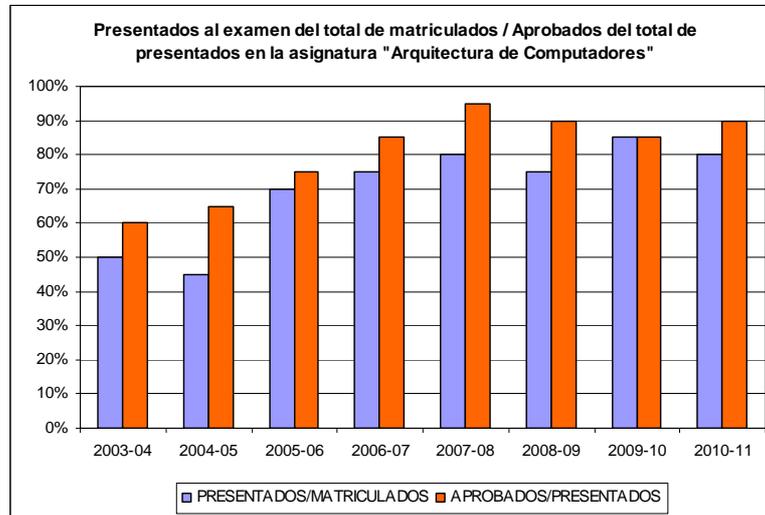


Figura 8. Gráfica sobre alumnos presentados a examen y sobre los aprobados desde el curso 2003-04. Estas metodologías se introdujeron en el curso 2005-2006.

5 Conclusiones

El aprendizaje cooperativo le permite al docente alcanzar varias metas importantes al mismo tiempo. En primer lugar, le ayuda a elevar el rendimiento de todos sus alumnos, incluidos tanto los especialmente dotados como los que tienen dificultades para aprender. En segundo lugar, lo ayuda a establecer relaciones positivas entre los alumnos, sentando así las bases de una comunidad de aprendizaje en la que se valore la diversidad. En tercer lugar, les proporciona a los alumnos las experiencias que necesitan para lograr un desarrollo social, psicológico y cognitivo. La posibilidad que brinda el aprendizaje cooperativo de abordar estos tres frentes al mismo tiempo lo hacen superior a todos los demás métodos de enseñanza. El uso de grupos de aprendizaje cooperativo ha mejorado no sólo los resultados académicos sino también la motivación hacia estas asignaturas, que de partida, encuentran más difíciles.

Uno de los puntos fuertes del *Portafolio* digital es la retroalimentación externa que el estudiante recibe continuamente de su trabajo, no sólo por parte del profesorado, sino por sus propios compañeros. Ello hace que el grupo reflexione sobre sus propios trabajos, su proceso de aprendizaje, sus resultados y sus conclusiones.

Los resultados obtenidos muestran que los estudiantes están más motivados y como consecuencia se implican activamente en las diferentes actividades propuestas, además de conseguir involucrar a un mayor número de alumnos en el seguimiento diario de la asignatura. Todo ello llevado al campo de los resultados evaluables, se ha traducido en un mayor número de presentados y también aprobados en la asignatura, así como en una mayor satisfacción, no sólo para los alumnos, sino también para los docentes que han impartido la materia.

Referencias

1. Panitz, T. (1999) *Benefits of cooperative learning in relation to student motivation*. En Michael The-all (Ed.). *Motivation from within: Approaches for encouraging faculty and students to excel*, New directions for Teaching and Learning, 78. San Francisco, CA: Josey-Bass publishers.
2. Johnson, D.W, Johnson, R.T. Holubec, E.J. (1999) *El aprendizaje cooperativo en el aula*. Ed. Paidós.
3. Klenowski, V. (2004) *Desarrollo de portafolio. Para el aprendizaje y la Evaluación*. Madrid: Narcea.
4. Gibbs, G. (2003) *Uso estratégico de la educación en el aprendizaje*. En Brown, S. y Glasner, A. (Ed.). *Evaluar en la Universidad. Problemas y nuevos enfoques* (pp. 61-75). Madrid: Narcea.
5. Barberà, E, Bautista, G.; Espasa, A., Guasch, T. (2006) *Portafolio electrónico: desarrollo de competencias profesionales en la red*. *Revista de Universidad y Sociedad del Conocimiento*, 3(2), pp. 55-66.
6. Johnson, D.W., Johnson, R.T. (1989). *Cooperation and competitions. Theory and research*. Edina, MN. Interaction Book Company.
7. Valero, M., Diaz de Cerio, L. (2005) *Autoevaluación y co-evaluación: estrategias para facilitar la evaluación continuada*. *Actas del I Simposio Nacional de Docencia en la Informática* (pp. 25-32), Granada, Septiembre.
8. Putnam, J. (1997) *Cooperative learning in diverse classrooms*. Upper Saddle River. N.J: Prentice-Hall.
9. Pujolàs, P. (2005) *El cómo, el porqué y el para qué del aprendizaje cooperativo*. *Cuadernos de Pedagogía*, 345, pp. 51-54.
10. Fabra, M.L. (1994). *Técnicas de grupo para la cooperación*. Barcelona: CEAC.
11. Villa, A., Poblete, M. (Dirs.) (2007) *Aprendizaje basado en competencias. Una propuesta para la evaluación de competencias genéricas*. Bilbao: Mensajero/ICE Universidad de Deusto.
12. Gil, C., Montoya, M.G., Herrada, R.I., Baños, R., Márquez, A.L., Alcayde, A. (2012) *Capítulo 7: Prácticas de aprendizaje cooperativo y del portafolio digital de grupo: Aprender a cooperar y cooperar para Aprender*, En M.F. Serrano (Coord). *Aprendizaje Cooperativo en Contextos Universitarios*, pp: 123-154. Editorial Universidad de Murcia.

Plan de Acción Tutorial para Alumno de Planes de Estudios a Extinguir en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación (TIPAT)

M. Rodríguez Álvarez¹; M. I. García Arenas¹; P. Paderewski Rodríguez,²; E. M. Ramos Ábalos³; J. García Miranda,⁴; M. A. Rubio Escudero⁵; N. Rico Castro³; P. A. Castillo Valdivieso¹.

¹*Departamento de Arquitectura y Tecnología de Computadores. E.T.S. Ingenierías Informática y de Telecomunicación. Universidad de Granada.*

²*Departamento de Lenguajes y Sistemas Informáticos. E.T.S. Ingenierías Informática y de Telecomunicación. Universidad de Granada.*

³*Departamento de Estadística e Investigación Operativa. Facultad de Ciencias. Universidad de Granada*

⁴*Departamento de Álgebra. E.T.S. Ingenierías Informática y de Telecomunicación. Universidad de Granada.*

⁵*Departamento de Ciencias de la Computación e Inteligencia Artificial. E.T.S. Ingenierías Informática y de Telecomunicación. Universidad de Granada.*

mgarenas@ugr.es, patricia@ugr.es, ramosa@ugr.es, jesusgm@ugr.es, mrubio@decsai.ugr.es, nrico@ugr.es, pedro@atc.ugr.es, manolo@ugr.es

Resumen. En este trabajo se presenta un Plan de Acción Tutorial (TIPAT) para alumnos de planes de estudio a extinguir en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación (ETSIIT) de la Universidad de Granada. TIPAT se ha venido desarrollando a lo largo de tres cursos académicos. Tras una introducción se presenta el Plan de Acción Tutorial, el desarrollo del trabajo, así como los resultados obtenidos y el beneficio que ha supuesto para los estudiantes la realización de este proyecto.

Palabras clave: Plan de acción tutorial (PAT), tutoría, orientación.

Abstract. This work shows a Tutorial Plan (named TIPAT) for students in extinct studies in the Faculty of Computer Science and Telecommunication Engineering of the University of Granada. TIPAT has been developed over three academic years. After an introduction this work presents the Tutorial Plan, development work, as well as the results and the benefits it has brought to students carrying out this project.

Keywords: Tutorial Plan, tutoring, mentoring.

1 Antecedentes

TIPAT nace con el objetivo de prestar ayuda a los estudiantes de planes de estudio a extinguir de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación (ETSIIT) de la Universidad de Granada que tienen pendientes asignaturas de las que no existen ya docencia presencial. Debido a las peculiares necesidades de estos estudiantes, se puso en marcha en el curso 2010-2011 este Plan de Acción Tutorial, conocido como TIPAT que ha continuado durante los cursos académicos 2011-2012 y 2012-2013. La propuesta original se centra en la acción tutorial hacia estudiantes que se encuentran matriculados de asignaturas que se han extinguido, dado que están cursando un plan de estudios que está siendo sustituido. Este problema tiene una duración temporal limitada; comienza en el curso 2010-2011 y se prolonga, al menos, durante los cuatro años en que progresivamente van suprimiéndose los cursos del plan antiguo.

En el curso 2012-2013 dejarán de impartirse las asignaturas de tercer curso de las tres titulaciones asociadas a los estudios de Informática y de la de los estudios de Ingeniero de Telecomunicación. Previamente, durante los cursos 2010-2011 y 2011-2012 dejaron de impartirse las asignaturas de primer y segundo curso. Esto hace que en las titulaciones de Ingeniero Técnico en Informática de Sistemas e Ingeniero Técnico en Informática de Gestión ya se haya extinguido totalmente el plan de estudios sin que se imparta docencia presencial. En las titulaciones de ciclo largo, a saber Ingeniero en Informática e Ingeniero de Telecomunicación todavía persiste la docencia presencial en los cursos 4º y 5º. La desaparición de la docencia presencial de estas asignaturas conlleva una pérdida general en el sistema de apoyo y ayuda que recibe el estudiante en comparación con la orientación habitual que se establece durante las clases presenciales.

El Proyecto TIPAT que se puso en marcha durante el curso 2010-2011 presenta, mediante distintas vías de actuación, una red de apoyo, información y orientación para los alumnos que se encuentran matriculados de asignaturas en cursos sin docencia. La continuación de este proyecto tiene por objeto ofrecer esta ayuda a los estudiantes que durante el curso 2012-2013 se van a encontrar sin docencia de asignaturas de tercer curso, y, posiblemente, también de primero y segundo.

Algunas cifras a tener en cuenta sobre las asignaturas que a extinguir en segundo curso son:

- Número de asignaturas que se extinguieron en el curso 2010-2011: **42**
- Número de matrículas realizadas en asignaturas sin docencia en el curso 2010-2011: **2177**
- Número de asignaturas que se extinguieron en el curso 2011-2012: **37**
- Número de matrículas realizadas en asignaturas sin docencia en el curso 2011-2012: **3136**
- Número de asignaturas sin docencia en el curso 2012-2013: **133**

Tabla 1. Asignaturas a extinguir. Curso 2012-2013.

Asignaturas a extinguir en 2012-2013			
Titulación	Primer cuatrimestre	Segundo cuatrimestre	Total
Ingeniero en Informática	14	14	28
I.T. Informática de Sistemas	21	19	40
I.T. Informática de Gestión	21	19	40
Ingeniero Telecomunicación	13	12	25
	69	64	133

El hecho de que desaparezca tan elevado número de asignaturas es debido a la gran oferta de asignaturas optativas que hay en tercer curso de las cuatro titulaciones.

Los Departamentos que tienen asignadas las asignaturas que desaparecen son:

- Álgebra
- Análisis Matemático.
- Arquitectura y Tecnología de Computadores.
- Ciencias de la Computación e Inteligencia Artificial.
- Economía Financiera y Contabilidad
- Electromagnetismo y Física de la Materia Condensada.
- Electrónica y Tecnología de Computadores.
- Estadística e Investigación Operativa.
- Física Aplicada.
- Lenguajes y Sistemas Informáticos.
- Matemática Aplicada.
- Organización de Empresas
- Teoría de la Señal, Telemática y Comunicaciones.

Dados los datos de los cursos 2010-2011 y 2011-2012, se puede estimar que para el curso 2012-2013 habrá más de 2500 matrículas en asignaturas sin docencia. El número de estudiantes que se ven afectados por esta situación puede ser muy elevado, más de 400 estudiantes. Por otra parte, en la puesta en marcha de TIPAT, durante los cursos 2010-2011 y 2011-2012 se recibieron más de 400 solicitudes de participación en cada curso, lo que sugiere que una gran parte de los estudiantes que se van a encontrar sin docencia presencial de tercer curso también demandarán la ayuda de TIPAT.

2 Descripción

En este Plan de Acción Tutorial, TIPAT, colaboran estudiantes, profesores y personal de administración y servicios vinculados con la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada. Se enmarca

dentro de los Proyectos de Innovación Docente que financia el Vicerrectorado de Ordenación Académica y Profesorado de la Universidad de Granada a través del Secretariado de Innovación Docente.

En la primera etapa de TIPAT, desarrollada durante el curso 2010-2011, formaron parte del proyecto un total de 345 estudiantes, 42 profesores y 2 trabajadores del área de Administración; en total 389 participantes vinculados con la Escuela Técnica de Ingenierías Informática y de Telecomunicación.

En su segunda etapa, desarrollada durante el curso 2011-2012, el número de participantes es de 203 (130 estudiantes, 71 profesores y 2 trabajadores del área de Administración). Además, en cada uno de los dos cursos académicos en los que TIPAT ha funcionado, ha contado con la ayuda técnica de un estudiante en prácticas, a través del Centro de Promoción de Empleo y Prácticas, del Vicerrectorado de Estudiantes de la Universidad de Granada.

3 Objetivos

Los objetivos de cualquier proyecto de estas características deben ser claros y no deben ser muchos para evitar abarcar más tareas de las que realmente se van a llevar a cabo. Además teniendo en cuenta que se trata de un proyecto de orientación y tutoría el objetivo general es claro: guiar a los alumnos en el plano personal, académico, social y profesional como una parte más de su proceso formativo. Teniendo en mente este primer objetivo general, se puede detallar y afinar más adecuando este objetivo general al ámbito de este proyecto. Y en este ámbito particular el objetivo final será orientar y colaborar con el alumnado en la preparación del temario de las distintas asignaturas sin docencia o a extinguir dentro del plan de estudios al que pertenezcan dichas asignaturas, persiguiendo el superar los exámenes de cada una de las asignaturas pendientes.

Detallando este objetivo general aún más, se pueden extraer una serie de objetivos específicos:

- Evaluar la situación de partida del alumno, para determinar sus conocimientos, debilidades, necesidades y potencialidades.
- Facilitar técnicas de estudio y de reciclaje de conocimientos.
- Informar de la actualización de los contenidos del temario así como de cambios en la forma, tipo o puntuación del examen.
- Crear un espacio de encuentro, en el que se transmitan los problemas y dificultades a los que se enfrentan los alumnos, y en el que se estimulen habilidades y actitudes positivas.
- Apoyar al alumno y realizar una labor de nexo de unión y vía de comunicación con las asignaturas extinguidas del plan antiguo.
- Potenciar el alumno activo y diseñar con él un itinerario de preparación de asignaturas.

4 Actividades Realizadas

Dentro del desarrollo de este proyecto se planificaron varias actividades a llevar a cabo a lo largo del curso académico, tanto para los estudiantes participantes como para los profesores tutores. En el caso de los profesores tutores, se impartió una charla inicial en la que se informaba del modo de actuación que se iba a seguir en el caso de que los alumnos que no atendieran a los tutores o en el caso de tutores que no atendieran debidamente a los alumnos. Además en esta charla se introdujeron las novedades con respecto a las actividades planificadas para los alumnos para que desde los primeros días de desarrollo del proyecto los profesores tutores tuvieran conocimiento de las actividades propuestas.

En el caso de los alumnos, se planificaron tres actividades a lo largo del todo el curso. La primera de ellas era una sesión informativa a nivel administrativo del procedimiento que se seguía para las asignaturas sin docencia, aclarando temas administrativos referentes al número de convocatorias disponibles por curso académico, o a lo largo de todos los años en los que el estudiante se hubiera matriculado de ella. Además se informó de la normativa vigente para exámenes, precios de matriculación, matriculas de asignaturas que constan como optativas en el plan de estudio a extinguir. La sesión fue impartida por el personal que colabora con el proyecto y que a la vez forma parte de la secretaria de la Escuela Superior de Ingeniería Informática y por lo tanto conoce a fondo los problemas que los estudiantes iban planteando. Esta sesión se desarrollo a modo de seminario y fue totalmente voluntario para los alumnos que quisieron asistir.

Por último, se organizaron dos sesiones informativas sobre las asignaturas que durante el curso académico 2011-2012 estaban sin docencia por primera vez. La primera de ellas se impartió en el primer cuatrimestre incluyendo las asignaturas cuya evaluación estaba prevista a final del primer cuatrimestre y la segunda para las asignaturas del segundo cuatrimestre. En ambos casos, sólo se impartieron charlas para las asignaturas que tenían un mínimo de personas matriculadas y cuyo profesor responsable de la evaluación estaba dispuesto a impartirla. Hay que destacar, que en todos los casos los profesores realizaron estas actividades de forma totalmente voluntaria y que en ningún caso hubo profesores que se negaran a hacerla después de que el grupo coordinador del proyecto se lo pidiera.

5 Resultados, Productos y Beneficios Generados por el Proyecto

La valoración de los resultados del proyecto se ha realizado a partir de un cuestionario de valoración por parte de los estudiantes, la creación de un buzón de sugerencias, la apertura de un foro de debate en la plataforma web y reuniones del equipo de coordinación.

5.1 Resultados de la evaluación del aprendizaje de los estudiantes

De los resultados de la encuesta se deduce como las sesiones de orientación, las actividades organizadas (charlas de presentación de asignaturas, talleres y sesiones de

reposito) y la utilización de la página web del proyecto (ver figura 1) han sido muy beneficiosas para el estudiante.

5.2 Resultados de la evaluación interna e instrumentos utilizados.

Los resultados de la encuesta han sido muy positivos, destacando que un 77.8 % del alumnado encuestado considera muy importante la orientación universitaria y un 83.3% que es muy importante la existencia de un plan de acción tutorial como TIPAT.



Fig. 1. Sitio web del plan de acción tutorial

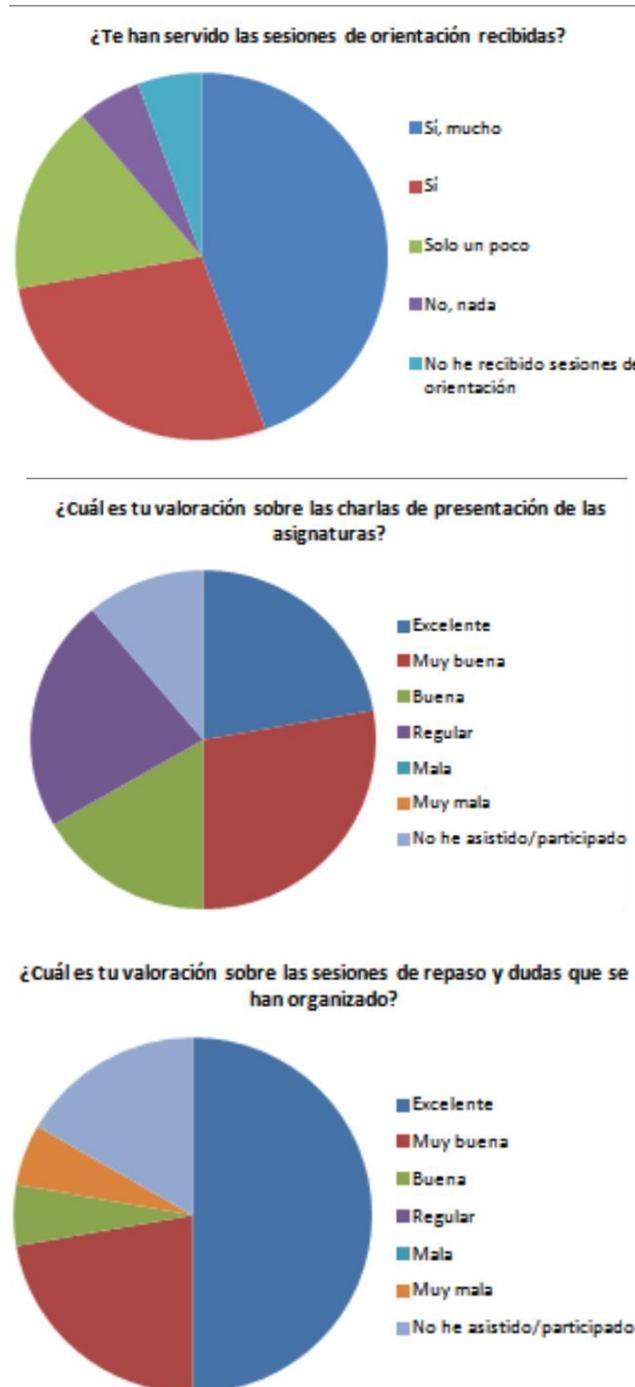


Fig. 2. Resultados de la evaluación del aprendizaje de los estudiantes

5.3 Resultados de la evaluación externa e instrumentos utilizados

Dado que la valoración general fue muy buena en todas las ocasiones en las que se presentó la primera edición de TIPAT en congresos, reuniones y seminarios a nivel nacional e internacional ([1], [2], [3] y [4]), se ha seguido la misma dinámica con la segunda edición de TIPAT, volviéndose a obtener una opinión muy positiva en el foro internacional presentado ([5]).

Para conocer la opinión del alumnado se ha creado en la plataforma virtual un buzón de sugerencias individualizado y privado, de forma que el alumno pueda expresar sus quejas, necesidades, impresiones y sugerencias en cualquier momento.

Además como modalidad complementaria al buzón de sugerencias, se ha abierto un foro de debate que permitirá visibilidad pública por parte de alumnos y profesores de los comentarios que se manden a la plataforma. De esta manera, las sugerencias planteadas serán moderadas entre todos, y el alumno tendrá una mejor percepción de su participación en la gestión de la experiencia.

6 Valoración Global

En general la puesta en marcha y el desarrollo del proyecto ha sido muy positiva, consiguiéndose todos los objetivos propuestos, y en particular para los alumnos tal y como muestran los resultados de la encuesta presentada en la figura 2.

Bibliografía

1. García Arenas, M.I.; Paderewski Rodríguez, P.; Ramos Ábalos, E.M.; García Miranda, J.; Rubio Escudero, M.A.; Rico Castro, N.; Castillo Valdivieso, P.A.; Rodríguez Álvarez, M.: "Implantación de un Plan de Acción Tutorial para alumnos de planes de estudio a extinguir de la ETSIT de la Universidad de Granada." II Jornadas sobre Innovación Docente y adaptación al Espacio Europeo de Educación Superior en las Titulaciones Técnicas. Universidad de Granada. Granada, 26 y 29 de Septiembre de 2011. ISBN 978-84-15418-00-9, pp. 419-422.
2. Paderewski Rodríguez, P.; García Miranda, J.; Rubio Escudero, M.A.; Cruz Corona, C.; Montes Soldado, R.A.; Rodríguez Álvarez, M.: "Plan de Acción Tutorial en la Universidad de Granada. Primeros resultados." II Congreso Internacional de Docencia Universitaria (CIDU 2011). Vigo (España), 30 de Junio a 2 de Julio de 2011.
3. Rico Castro, N.; Castillo Valdivieso, P.A.; Rubio Escudero, M.A.; Paderewski Rodríguez, P.; García Arenas, M.I.; Ramos Ábalos, E.M.; García Miranda, J.; Rodríguez Álvarez, M.: "A new tutorial system for computer science and telecommunication engineering students" Congreso New Perspectives in Science Education. Florencia (Italia), 8 y 9 de Marzo de 2012.
4. Rico Castro, N.; García Arenas, M.I.; Ramos Ábalos, E.M.; Rubio Escudero, M.A.; Romero García, S.F.; Rodríguez Álvarez, M.: "Plan de acción tutorial para alumnos de planes de estudio a extinguir en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación." Actas de las I Jornadas sobre Inno-

vacación Docente y Adaptación al EEES en las Titulaciones Técnicas. Granada, 9 y 10 de Septiembre de 2010, pp. 215 a 218.

5. Díez Sánchez, M.A.; Rico Castro, N.; García Arenas, M.I.; Paderewski Rodríguez, P.; Castillo Valdivieso, P.A.; Rodríguez Álvarez, M.: “Sistema web de apoyo para el desarrollo de un Plan de Acción Tutorial.” III Jornadas sobre Innovación Docente y adaptación al Espacio Europeo de Educación Superior en las Titulaciones Técnicas. Universidad de Granada. Granada, 20 y 21 de Septiembre de 2012. ISBN 978-84-15418-73-3, pp. 203-207.

Estudio sobre algoritmos genéticos en la nube y el modelo de programación MapReduce*

G. Muñoz, P. García-Sánchez, P. A. Castillo, M. G. Arenas, A. M. Mora, y J. J. Merelo¹

Dpto. de Arquitectura y Tecnología de los Computadores, Universidad de Granada
munozs.88@gmail.com, {pgarcia, pedro, mgarenas}@atc.ugr.es, {amorag,
jmerelo}@geneura.ugr.es

Resumen Este trabajo presenta el proyecto fin de carrera “Estudio sobre algoritmos genéticos en la nube y el modelo de programación MapReduce”. Durante el desarrollo de este proyecto se investigó en el uso y aplicación de Algoritmos Genéticos en distintos entornos de Cloud Computing, como el MapReduce o virtualización de instancias. Se ejecutaron distintas configuraciones de parámetros del algoritmo (como el tamaño de población o el tipo de crossover) en distintas instancias de Amazon Web Services. Los resultados muestran el efecto de estos parámetros al tipo de instancia utilizada.

Palabras clave Cloud Computing, Proyecto Fin de Carrera, Amazon EC2, Virtualización, MapReduce, Algoritmos Evolutivos

Abstract This paper shows the final degree project “A study of genetic algorithms in the cloud and the MapReduce model”. During the development of this project the usage and application of genetic algorithms in different Cloud Computing environments was investigated, such as MapReduce or virtualization. Different parameter configurations, such as the population size or crossover type, were launched in different instances of Amazon Web Services. Results show the effect of these parameters to the different types of used instances.

Keywords Cloud Computing, Final Degree Project, Amazon Ec2, Virtualization, MapReduce, Evolutionary Algorithms

1. Introducción

En la última década Internet ha transformado nuestra economía y nuestra sociedad. Ha demostrado ser una infraestructura de comunicación y enlace de gran valor que se adapta gradualmente a las necesidades de los usuarios. Con Internet

* Financiado con los proyectos EvOrq (TIC-3903), Beca FPU AP2009-2942, ANY-SELF (TIN2011-28627-C04-02) y CANUBE (Proyecto 83 CEI-BIOTIC).

se ha creado una red mundial de intercambio de conocimientos, creatividad y colaboración.

Del mismo modo, Internet ha dado un gran salto adelante gracias al despliegue de la banda ancha de alta velocidad, lo que ha permitido el lanzamiento de muchos nuevos servicios interactivos y de contenido.

Todo esto hace que los programas informáticos ofertados en la red como un servicio disminuyan sus costes y aumenten su eficacia, provocando una gran mejora de la productividad. Desplegado adecuadamente, la llamada Internet del futuro, que cada vez es más presente, trae consigo innovación, aumento de la productividad, nuevos mercados y más crecimiento y empleo en esta década que recién acabamos de comenzar.

En este marco se engloba el Cloud Computing, un nuevo modelo de negocio en Internet, que beneficia tanto al proveedor de servicios en la nube como al usuario, capaz de tener acceso instantáneo a una infraestructura escalable y potente a un coste mínimo.

Paralelamente a esta transformación, la ciencia ha estado evolucionando constantemente durante los últimos lustros, en gran parte, gracias a la aparición y el uso de nuevas tecnologías. La investigación científica avanza al unísono con ésta, pues un cambio tecnológico, la mayoría de las veces, abre un nuevo abanico de posibilidades para la comunidad científica ya sea para el desarrollo en un ámbito clínico, científico o industrial, ya que la necesidad de capacidad de procesamiento de datos es común al sector en el que se desarrollen.

Por otra parte, el crecimiento de los volúmenes de datos a procesar es exponencial, por lo que para resolver un problema complejo el personal científico o investigador tiene que tratar con una cantidad de información ingente. Dentro de estos conjuntos de problemas se encuentran los de búsqueda, a menudo abordados por los denominados Algoritmos Genéticos, que se usan cada vez más en escenarios de gran complejidad.

Sentando estos principios en la sociedad actual, la motivación de este proyecto fin de carrera fué el de buscar una aproximación nueva para desarrollar trabajos de investigación usando las facilidades que nos brinda el Cloud Computing, y concretamente la resolución de problemas de búsqueda representados mediante Algoritmos Genéticos, yendo más allá y sentando las bases de un framework como *Hadoop Map Reduce*, para la reducción de complejidad y tiempo de procesamiento en los métodos de búsqueda.

El resto del trabajo se organiza como sigue: primero se define el concepto de Cloud Computing en la Sección 2. A continuación se explican dos tipos de tecnologías a aplicar los experimentos: *Amazon Web Services* y el modelo *Map-Reduce* y (en las Secciones 3 y 4 , respectivamente). Los Algoritmos genéticos se presentan en la Sección 5. Después se describen los experimentos a realizar (Sección 6) y el análisis de sus resultados (Sección 7). Finalmente se presentan las conclusiones y trabajo futuro.

2. Cloud Computing

Cloud computing es aún un paradigma en evolución, sus definiciones, arquitecturas, modelos, casos de uso, tecnologías base, problemas, riesgos y beneficios, son continuamente redefinidos en debates promocionados por el sector público y privado. En [1] podemos ver varias definiciones aunque muchos citan como más precisa la definición dada por el *National Institute of Standards and Technology* (NIST), donde se define el *Cloud Computing* como:

Cloud Computing es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables, por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente proporcionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios. Este modelo de nube promueve la disponibilidad y está compuesto por cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue.

En un intento de adentrarse en lo que es y no es el Cloud computing compararemos este paradigma con la computación tradicional. Con la computación tradicional ejecutamos copias de software en cada ordenador. Los documentos que creamos son almacenados en el ordenador en el que fueron creados. Aunque se puede acceder a los documentos desde otros ordenadores de la red, no se puede desde ordenadores de fuera de ella.

Con cloud computing, los programas que usamos no se ejecutan en nuestro ordenador personal, sino que son almacenados en servidores accedidos vía Internet. Si tu ordenador se rompiera, el software seguiría estando operativo para los demás. De la misma forma, los documentos que creamos son almacenados en una colección de servidores en los que un usuario con permiso puede, no sólo acceder a los documentos, sino también editar y colaborar en ellos en tiempo real. A diferencia de la computación tradicional, el modelo de Cloud no es PC-Céntrico por lo que el dispositivo que se usa para acceder a la información no es lo más importante.

A continuación, veremos qué es Cloud Computing y qué no es [2] .

2.1. Qué no es cloud computing

Cloud computing no es una red de ordenadores. Con una red, las aplicaciones y documentos son almacenados en un servidor de la empresa y accedidos a través de la red de la empresa. Cloud computing es mucho más que esto. Engloba múltiples empresas, múltiples servidores y múltiples redes. A diferencia de una red de ordenadores, los servicios cloud y el almacenamiento pueden ser accedidos desde cualquier ordenador del mundo con una conexión a internet.

Cloud computing tampoco es la subcontratación de un servicio externo de manera tradicional, donde la compañía contratada presta sus servicios a la empresa contratadora. Mientras una empresa de outsourcing organiza sus datos y aplicaciones, estos documentos y programas solamente son accesibles a los empleados de la empresa a través de la red de la misma, no a todo el mundo vía

internet. Así que, a pesar de las aparentes similitudes, las redes y el outsourcing no son cloud computing.

2.2. Qué es cloud computing

La clave de la definición de Cloud Computing es el Cloud en sí mismo. Para nuestro propósito, el cloud es un gran conjunto de ordenadores interconectados. Estos computadores pueden ser PC's o servidores, públicos o privados.

Esta nube de computadores se extiende más allá de una simple compañía o empresa. Las aplicaciones y datos servidos por la nube están disponibles para un amplio grupo de usuarios, a través de empresas y plataformas. El acceso es vía Internet. Cualquier usuario autorizado puede acceder a estos documentos y aplicaciones desde cualquier ordenador con conexión a Internet. Y, para el usuario, la tecnología e infraestructura que se esconde detrás de la nube es invisible.

Para ejemplificar algunos de los puntos más interesantes del cloud computing nos basaremos en Google, una de las principales compañías impulsoras del Cloud. Según su perspectiva, hay seis propiedades clave en el cloud computing.

- Es “Centrado en el usuario”: Una vez que tú eres un usuario conectado a la nube, cualquier cosa que haya almacenada allí - documentos, mensajes, imágenes, aplicaciones - se convierte en tuya. Además, no solo los datos son tuyos sino que puedes compartirlos con otros. En efecto, cualquier dispositivo que acceda a tus datos en la nube también convierte los datos en suyos.
- Es “Centrado en la tarea”: En vez de concentrarse en la aplicación y qué hacer, se centra en lo que necesitas y cómo la aplicación puede hacerlo por ti. Las aplicaciones tradicionales - procesadores de textos, hojas de cálculo, email... - se están convirtiendo en más importantes que los documentos que se crean.
- Es poderoso: Conectando cientos o miles de computadores juntos en una nube se crea una riqueza de computación imposible de hacer en un único PC.
- Es accesible: Como los datos son almacenados en la nube, usuarios pueden recuperar información instantáneamente desde múltiples repositorios. No estás limitado a una única fuente de datos, como lo estás en un PC.
- Es inteligente: Con el volumen de datos almacenados en la nube, el data-mining y análisis son necesarios para acceder a esta información de una manera inteligente.
- Es programable: Muchas de las tareas necesarias para el cloud computing deben de ser automatizadas. Por ejemplo, para proteger la identidad de los datos, la información almacenada en un único computador en la nube debe ser replicada en los otros computadores de la nube. Si uno de éstos falla, la programación de la nube redistribuirá automáticamente los datos de éste computador hacia uno nuevo de la nube.

3. Amazon Web Services

En esta sección se presenta *Amazon Web Services* [3] en general, y en particular, los servicios que se han utilizado para el desarrollo de este trabajo.

3.1. Características

Amazon EC2 proporciona un servicio Web que permite obtener y configurar recursos computacionales de forma sencilla. Proporciona un control completo de los recursos computacionales y permite al usuario emplear el entorno de Amazon. Amazon EC2 reduce el tiempo requerido para obtener e iniciar nuevas instancias a minutos, permitiendo escalar la capacidad de forma rápida y en función de los requerimientos. Amazon EC2 permite al usuario adquirir la capacidad que en cada momento necesita. Además, proporciona a los desarrolladores las herramientas para construir aplicaciones robustas y aislarlas de los escenarios comunes propensos a fallos.

Amazon EC2 es un entorno de computación virtual que permite emplear interfaces Web para lanzar instancias de una gran variedad de sistemas operativos, instalar entornos de aplicaciones personalizados, gestionar los permisos de acceso a la red, y ejecutar las imágenes virtuales usando tantos sistemas como sea preciso.

Para emplear Amazon EC2 es necesario:

- Seleccionar una plantilla preconfigurada para ser ejecutada de forma inmediata, o crear una Amazon Machine Image (AMI) que contenga sus aplicaciones, librerías, datos y las opciones de configuración asociadas.
- Configurar la seguridad y el acceso a la red para la instancia de Amazon EC2.
- Escoger el tipo de instancia y el sistema operativo que desee y, a continuación, iniciar, apagar y monitorizar tantas instancias como sea necesario, usando las APIs del servicio Web o la gran variedad de herramientas proporcionadas.
- Determinar si se ejecutarán las máquinas virtuales en múltiples ubicaciones, si se utilizarán direcciones IP estáticas o se conectarán sistemas de almacenamiento persistente a las mismas.
- Pagar sólo por los recursos que se consumen, como el de tiempo de CPU de las instancias o los datos transferidos.

3.2. Aspectos destacados del servicio

- **Elástico.** Amazon EC2 permite incrementar o reducir la capacidad en minutos. Se podría ordenar la iniciación de cientos o miles de instancias de máquinas virtuales de forma simultánea. La aplicación de usuario podría escalar de forma automática los recursos necesarios en función de las necesidades de la aplicación empleando el API del servicio Web de Amazon.
- **Controlado por completo.** Se tiene el control completo de las instancias. El usuario tiene acceso de root a cada una de ellas y, por tanto, podría controlarlas de la misma forma que cualquier otra máquina.

- **Flexible.** Permite seleccionar distintos tipos de instancias, sistemas operativos y paquetes de software. Amazon EC2 permite seleccionar la configuración de memoria, CPU, espacio de almacenamiento, tamaño de la partición de inicio.
- **Diseñado para ser empleado con otros servicios Web de Amazon.** Amazon EC2 emplea el servicio de almacenamiento Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB y Amazon Simple Queue Service (Amazon SQS) para proporcionar una solución completa a la computación, el procesamiento de colas y el almacenamiento en un amplio rango de aplicaciones.
- **Confiable.** Amazon EC2 ofrece un entorno confiable donde puedan reemplazarse las instancias de forma rápida y predecible. El acuerdo de nivel de servicio de Amazon es del 99,95 % para cada una de las zonas de la región de Amazon EC2.
- **Seguro.** Amazon EC2 proporciona numerosos mecanismos de seguridad para los recursos. Incluye interfaces Web para configurar las opciones de firewall entre sus grupos de instancias.
- **Barato.** Se cobra una tasa reducida por la capacidad de computación que se consume. Existen diferentes servicios:
 - *Instancias bajo demanda.* Permiten pagar por la capacidad de computación por cada hora de utilización sin la necesidad de establecer compromisos a largo plazo. Esto permite liberar al usuario de los costes y la complejidad que supone la planificación, la adquisición y el mantenimiento de hardware.
 - *Instancias reservadas.* Las instancias reservadas permiten efectuar un pago inicial bajo, de una sola vez para cada instancia que desea reservar y disponer de una tasa de descuento significativo en la utilización por hora de la misma.
 - *Instancias Spot.* Permiten al usuario efectuar una oferta por la capacidad no utilizada de Amazon EC2, lo que permitiría ejecutar las instancias durante el tiempo que el precio ofertado supere al precio actual de los recursos no utilizados.

4. Modelo MapReduce

Durante años, los programadores se han visto forzados a realizar implementaciones específicas para trabajar con grandes volúmenes de datos en sus entornos distribuidos de trabajo.

Pese a que no siempre es necesario, el tamaño de la información entrante suele ser grande y los cálculos deben ser distribuidos entre una gran multitud de máquinas para acabar en un tiempo razonable. Los problemas de cómo paralelizar estos cálculos, distribuir la información y manejar los fallos dificultan bastante la labor de implementación.

El modelo de programación MapReduce es una propuesta que pretende resolver las dificultades anteriores, ya que sus características y motivación se basan en la delegación de los cómputos intensivos en datos a un clúster de máquinas remotas que, mediante un sistema de ficheros distribuido, repartirán la carga de

trabajo, optimizando tiempo y recursos. Asimismo, facilita un patrón de desarrollo paralelo para simplificar la implementación de aplicaciones intensivas en datos en entornos distribuidos. Este modelo puede dividir un espacio grande de problema en espacios pequeños y paralelizar la ejecución de tareas más pequeñas en estos sub-espacios.

Este modelo ha cobrado especial interés por su aplicabilidad en entornos de Cloud Computing [2]. MapReduce ofrece unas ventajas muy directas y evidentes como es la centralización de los datos en servidores remotos, eliminando las dependencias con los soportes físicos; o la contratación de servicios en función de las necesidades de las empresas, sin tener que añadir equipos, software o personal, lo que conlleva un considerable ahorro también en el plano energético.

MapReduce [4] es un modelo de programación, desarrollado por Google, que es utilizado para procesar grandes conjuntos de datos distribuidos a lo largo de un clúster de servidores. Este procesado computacional puede tener lugar tanto sobre datos almacenados en sistemas de ficheros, como en bases de datos. El modelo de programación está inspirado en los lenguajes funcionales y permite al desarrollador expresar sus algoritmos utilizando únicamente dos funciones, **map** y **reduce**.

Las funciones map y reduce de MapReduce se definen sobre datos estructurados en pares clave-valor. La función map, escrita por el usuario, recibe un par clave-valor y devuelve un conjunto de pares clave-valor intermedio:

$$\text{map} : (k1, v1) \rightarrow [(k2, v2)] \quad (1)$$

Esta función (1) se aplica en paralelo a cada par del conjunto de datos de entrada produciendo una lista de pares $(k2, v2)$ por cada llamada. MapReduce agrupa todos los valores intermedios asociados con la misma clave k y se los pasa a la función reduce.

La función reduce (2) recibe dicha clave y su conjunto de valores asociados y los fusiona para formar un conjunto de valores más pequeño:

$$\text{reduce} : (k2, [v2]) \rightarrow [v3] \quad (2)$$

Cada llamada reduce produce bien una lista $v3$ o un valor vacío. Los resultados de las llamadas se recopilan en la lista de resultados buscada.

Desde la perspectiva del flujo de datos, la ejecución de MapReduce consiste en M tareas map y R tareas reduce independientes. Generalmente, los resultados intermedios se particionan en R trozos para R tareas reduce.

Los principales elementos de un flujo de trabajo MapReduce son los siguientes:

- **Proceso que lee la entrada:** divide los datos de entrada en bloques, siendo asignados cada uno de estos bloques a la función map correspondiente. Estos datos serán leídos de un almacenamiento estable y generará pares clave/valor. Un ejemplo común sería la lectura de un directorio entero, y la devolución de un registro por cada línea de cada fichero.

- **Función de mapeo (map):** Cada función map recibe una serie de pares clave-valor, los procesa individualmente, y devuelve cero o más pares clave-valor de salida. Los tipos de datos de la entrada y la salida de la función map suele ser distintos. Por ejemplo, si la aplicación realizase un conteo de palabras, la función map separaría la línea en palabras y sacaría como resultado la palabra como clave y un 1 como valor.
- **Función de partición:** Las salidas de cada uno de los nodos map son asignadas a un nodo reduce en concreto a partir del resultado obtenido por la función “partition” de la aplicación. Esta función devuelve el índice del reduce buscado, dada una clave y un número de nodo reduce. Una función típica es hallar el valor hash de la clave y hacer módulo del número de nodo reduce.
- **Función de comparación:** La entrada de cada reduce se obtiene de la máquina en la que se ejecutó y ordenó el map utilizando la función de comparación de la aplicación.
- **Función de escritura de salida:** Escribe la salida de la función reduce en un almacenamiento estable, típicamente un sistema de ficheros distribuido.

El principal beneficio de este modelo de programación es la simplicidad. El programador simplemente proporciona una descripción del algoritmo centrada en su funcionalidad.

5. Algoritmos Genéticos

La técnica de búsqueda conocida como Algoritmo Genético se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes de un individuo (unidad básica de codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables (es decir, los que le permiten adaptarse mejor a su entorno) se transmiten a sus descendientes cuando éste se reproduce sexualmente.

La primera mención del término Algoritmo Genético, y la primera publicación sobre una aplicación del mismo, se deben a J.D.Bagley [5], que diseñó AGs para buscar conjuntos de parámetros en funciones de evaluación de juegos. Pero es otro científico el considerado creador de los AGs: John Holland, que los desarrolló, junto a sus alumnos y colegas, durante los 60 y 70.

El propósito original de Holland no era diseñar algoritmos para resolver problemas concretos, sino estudiar, de un modo formal, el fenómeno de la adaptación tal y como ocurre en la naturaleza y desarrollar vías de extrapolar esos mecanismos de adaptación natural a los sistemas computacionales.

En [6] Holland presentaba el AG como una abstracción de la evolución biológica, y proporcionaba el entramado teórico para la adaptación bajo el algoritmo genético. El AG de Holland era un método para desplazarse, de una población de cromosomas a una nueva población, utilizando un sistema similar a la selección

natural junto con los operadores de cruce, mutación e inversión inspirados en la genética. En este primitivo algoritmo, cada cromosoma consta de genes (bits) y cada uno de ellos es una muestra de un alelo particular (0 ó 1). El operador de selección escoge entre los cromosomas de la población aquellos con capacidad de reproducción, y entre éstos, los que sean más compatibles producirán más descendencia que el resto. El operador de cruce extrae partes de dos cromosomas, imitando la combinación biológica de dos cromosomas aislados (gametos). Y por último, el operador de mutación se encarga de cambiar, de modo aleatorio, los valores del alelo en algunas localizaciones del cromosoma.

5.1. Algoritmo Genético Básico

En esta sección se describen los componentes de un algoritmo genético básico y sus formas más comunes.

Representación de los Individuos Desde los primeros estudios de los AGs los individuos son cadenas binarias, siendo hoy en día aún la aproximación más utilizada, aunque existen otras que utilizan letras o valores numéricos para representar a sus cromosomas.

La sencillez de la representación binaria que utilizan los AGs les aporta características muy importantes de eficiencia. Sin embargo debemos disponer de un método para poder evaluar la adecuación del individuo como solución al problema. Lógicamente, el método de transformación es específico del problema considerado. Sin embargo, a la hora de diseñar el método de codificación es importante tener en cuenta una serie de directrices. Así, debemos buscar una codificación tal que cada punto del espacio de búsqueda esté representado por el mismo número de cadenas binarias, y tal que sea capaz de representar todos los puntos del espacio del problema.

En el contexto de los AGs, el término cromosoma se refiere a un candidato a solución del problema, que frecuentemente es codificado como una cadena de bits. Los genes son tanto un bit como bloques cortos de bits adyacentes que codifican un elemento particular del candidato a solución. Un alelo en una cadena de bits será un 0 o un 1 (para alfabetos largos cada posición puede tener más alelos). El genotipo de un individuo en un AG que emplea cadenas de bits es, simplemente, la configuración de bits del cromosoma de ese individuo.

Cabe resaltar que en este trabajo se usó tanto codificación binaria como codificación real que consiste en exactamente lo mismo pero un gen es representado como un número real, generalmente en un dominio dado por el problema.

Funcionamiento Sea X el problema a resolver, el esquema general de un algoritmo genético es el siguiente [7]:

A continuación se procederá a una explicación exhaustiva de dicho algoritmo. Dada una representación de candidatos a soluciones en una cadena de bits, un AG simple, tal y como se describe en [8], funcionaría del siguiente modo:

1. Comenzar con una población P generada aleatoriamente de n cromosomas de L bits.
2. Calcular el valor de la función de evaluación o fitness ($f(x)$) para cada cromosoma x de P.
3. Repetir los siguientes pasos hasta que se hayan creado todos los descendientes:
 - a) Seleccionar un par de cromosomas de P, siendo la probabilidad de selección proporcional al fitness. Los cromosomas seleccionados serán llamados cromosomas padre.
 - b) Con probabilidad p_c (tasa de cruce), cruzar el par de cromosomas padre en un punto (o más) elegido aleatoriamente para formar dos descendientes. Si no tiene lugar ningún cruce, formar dos descendientes que sean copias exactas de sus respectivos padres.
 - c) Mutar los dos descendientes en cada lugar con probabilidad p_m (tasa de mutación), y colocar los cromosomas resultantes en la nueva población P'.
4. Reemplazar la población actual P con la nueva población P'.
5. Evaluar la condición de finalización.
6. Volver al paso 2.

En el paso 2 del algoritmo se habla de una función de evaluación, la cual debe ser diseñada para cada problema de manera específica. Dado un cromosoma cualquiera, la función de evaluación le asigna un número real, que se supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

Asimismo, la primera operación del paso 3 del algoritmo es la llamada fase de selección. Esta selección se efectúa usando un procedimiento que favorezca a los individuos mejor adaptados. Existen diferentes métodos de selección, que posteriormente veremos.

Cada iteración de este proceso recibe el nombre de generación. Cada generación se obtiene a partir de la anterior por medio de los llamados operadores de reproducción (paso 3 del algoritmo), que pueden ser de dos tipos:

1. **Copia:** Es un tipo de reproducción asexual, en la que un determinado número de individuos pasa directamente a la siguiente generación, sin sufrir ningún proceso de variación en sus genes.
2. **Cruce:** Es una reproducción de tipo sexual en la que se genera una descendencia a partir de un número fijo de individuos, dos por lo general, de la generación anterior. Existen varios tipos de cruce, que veremos más tarde.

El algoritmo acabará cuando se cumpla la condición de fin, que normalmente será una de las siguientes:

1. Se ha encontrado una solución que satisface un criterio mínimo.
2. Se ha llegado a un número determinado de generaciones establecido previamente.
3. Se ha llegado a un límite preestablecido en tiempo de computación.
4. Tras varias generaciones el fitness de la mejor solución no ha variado.

5. Parada manual.
6. Combinación de las anteriores

Posiblemente el criterio más utilizado sea el primero, según el cual De Jong en su tesis doctoral [9] afirmó que si el AG es correcto, la población evolucionará a lo largo de las sucesivas generaciones de tal forma que la evaluación media entre todos los individuos, así como la propia del mejor individuo, convergerán hacia el óptimo global. Se dice que un gen ha convergido cuando al menos el 95 % de los individuos de la población comparten el mismo valor para dicho gen. Se dice que la población converge cuando todos los genes han convergido. Se puede generalizar dicha definición al caso en que al menos un β % de los individuos de la población hayan convergido.

El esqueleto de este algoritmo es la base de la mayoría de las aplicaciones de los algoritmos genéticos. Se podría profundizar mucho más en detalles sobre cuales deben ser las diferentes probabilidades, tamaño de la población y número de generaciones. De esos detalles dependerá, en gran parte, el éxito o fracaso de nuestro algoritmo.

6. Experimentos

Esta sección muestra los experimentos a realizar para hacer comparativas sobre la mejor configuración para AGs en Amazon EC2.

6.1. Parámetros de los algoritmos

Se van a usar diferentes configuraciones para ver el comportamiento y tiempo de ejecución en los distintos entornos. Primero se explicará lo que significa cada columna de las tablas y a continuación se mostrarán.

- **Función:** Función a optimizar, ya sea maximizar o minimizar.
- **Población:** Tamaño de la población de individuos.
- **Individuo:** Longitud de cada individuo en bits.
- **Generaciones:** Número de generaciones que queremos evaluar.
- **Mutrate:** Tasa de mutación en %.
- **Crossrate:** Tasa de cruce en %.
- **Selrate:** Tasa de selección en %.
- **Selección:** Método de selección.
- **Cruce:** Método de cruce.
- **Mutación:** Método de mutación.
- **Reemplazamiento:** Método de reemplazamiento.

Todos los parámetros y configuraciones se han establecido buscando la mayor variedad posible con el objeto de tener diferentes opiniones sobre la ventajas e inconvenientes en escalabilidad, tiempo de ejecución...

Configuración	Población	Bits	Generaciones	Selrate	Crossrate	Mutrate
Config.1	10000	512	100	30	10	10
Config.2	30000	512	200	20	20	20
Config.3	50000	512	500	40	30	10

Cuadro 1. Rastrigin. Tabla de configuraciones establecidas.

Configuración	Selección	Cruce	Mutación	Reemplazamiento
Config.1	Probabilistic (10%)	One-Point	NRandom (1)	Worst
Config.2	Deterministic	Two-Point	NRandom (1)	Worst
Config.3	Probabilistic (10%)	One-Point	NRandom (1)	Worst

Cuadro 2. Rastrigin. Metodos de selección, cruce, mutación y reemplazamiento.

Entorno Local La solución secuencial fue probada en una máquina aislada, que corría Ubuntu 12.04 de 64 bits. El equipo en cuestión es un HP Pavilion DV6 3034ss, Intel(R) Core(TM) i5 CPU M 450 @ 2.40GHz, 4 GB RAM.

Entorno Cloud de Amazon La batería de ejecuciones en la nube de Amazon consta de diferentes máquinas virtuales, instancias con diferentes características cada una:

- **Micro:** 613 MB de memoria, un máximo de dos unidades informáticas EC2 (para ráfagas periódicas cortas) y únicamente almacenamiento EBS. Plataforma de 32 o 64 bits.
- **Small:** Dispone de 1.7 GB de memoria, una unidad de computación EC2 (1 núcleo virtual con una unidad informática EC2), 160 GB de almacenamiento, plataforma de 32 o 64 bits.
- **Medium:** Dispone de 3.75 GB de memoria, dos unidades de computación EC2 (1 núcleo virtual con dos unidades informáticas EC2), 410 GB de almacenamiento, plataforma de 32 o 64 bits.
- **Large:** Dispone de 7.5 GB de memoria, cuatro unidades de computación EC2 (2 núcleos virtuales con dos unidades informáticas cada uno de ellos), 850 GB de almacenamiento, plataforma de 64 bits.
- **ExtraLarge:** Dispone de 15 GB de memoria, ocho unidades de computación EC2 (4 cores virtuales con dos unidades informáticas cada uno de ellos), 1690 GB de almacenamiento, plataforma de 64 bits.

Configuración	Población	Individuo	Generaciones	Selrate	Crossrate	Mutrate
Config.1	2500	64	800	40	15	10
Config.2	10000	64	1000	20	30	10
Config.3	50000	64	2000	40	15	10

Cuadro 3. Marea. Tabla de configuraciones establecidas.

Configuración	Selección	Cruce	Mutación	Reemplazamiento
Config.1	Probabilistic (15 %)	Two-Point	NRandom (1)	Generational
Config.2	Deterministic	One-Point	NRandom (2)	Worst
Config.3	Probabilistic (15 %)	Two-Point	NRandom (1)	Generational

Cuadro 4. Marea. Metodos de selección, cruce, mutación y reemplazamiento.

Configuración	Población	Individuo	Generaciones	Selrate	Crossrate	Mutrate
Config.1	10000	512	1000	30	30	10
Config.2	20000	512	3000	50	30	20
Config.3	40000	512	3000	50	30	10

Cuadro 5. OneMax. Tabla de configuraciones establecidas.

- **High-CPU ExtraLarge:** 7 GB de memoria, 20 unidades informáticas EC2 (8 núcleos virtuales con 2,5 unidades informáticas EC2 cada uno), 1.690 GB de almacenamiento de instancias, plataforma de 64 bits.

Los resultados de todas las ejecuciones para cada tipo de instancia, pueden consultarse en la dirección [https://s3.amazonaws.com/secuencial/results/\\$<Problema>\\$\\$<Configuracin>\\$.txt](https://s3.amazonaws.com/secuencial/results/$<Problema>$$<Configuracin>$.txt)

siendo <Problema> el problema que se desea consultar y <Configuración> el número de configuración. Así, por ejemplo, si se desea consultar los resultados del problema “Marea” con la configuración 2, habría que descargar el fichero desde: <https://s3.amazonaws.com/secuencial/results/Marea2.txt>

Veamos, primero, las tablas de resultados de tiempo de ejecución, para cada problema y tanto en entorno local, como en entorno Cloud, en cada tipo de instancia. Podemos apreciarlos en las tablas

La convergencia de los algoritmos puede ser vista en los ficheros de resultados antes nombrados. No obstante, vamos a estudiar algún caso para verla gráficamente.

Se puede observar que el portatil, el llamado entorno local, se ha comportado muy bien, puesto que tiene tiempos de manera similar a la instancia ExtraLarge y HCPUEXtraLarge, por lo que en el caso de querer hacer pocas pruebas, quizás con el entorno local valdría, pero no es el caso, porque hemos hecho una batería relativamente grande de pruebas.

Por otra parte, las instancias Small y Micro, quedarían totalmente descartadas para algún futuro proyecto, puesto que el tiempo excede de los límites permitidos, sobre todo la micro instancia. Estas instancias, pueden ser válidas

Configuración	Selección	Cruce	Mutación	Reemplazamiento
Config.1	Deterministic	Two-Point	NRandom (1)	Worst
Config.2	Probabilistic (15 %)	One-Point	NRandom (2)	Generational
Config.3	Probabilistic (15 %)	One-Point	NRandom (1)	Worst

Cuadro 6. OneMax. Metodos de selección, cruce, mutación y reemplazamiento.

Configuración	Local	Micro	Small	Medium	Large	ExtraLarge	High CPU EL
Config. 1	62	225	117	62	72	58	54
Config. 2	354	Error	703	373	424	338	320
Config. 3	1972	Error	Error	1583	1807	1434	1357

Cuadro 7. Tiempo de Ejecución en segundos. Rastrigin.

Configuración	Local	Micro	Small	Medium	Large	ExtraLarge	High CPU EL
Config. 1	16	47	41	21	24	19	17
Config. 2	164	701	425	222	253	203	169
Config. 3	815	4067	2070	1070	1215	974	818

Cuadro 8. Tiempo de Ejecución en segundos. Marea.

en el Cloud Computing como servidor de pequeñas webs o pruebas que no requieran de demasiado cómputo de datos.

Pasemos ahora a estudiar la escalabilidad, que viendo los ficheros de resultados queda demostrada que existe, pero el objetivo es ver el comportamiento según la configuración y los métodos escogidos. Se ha escogido como instancia representativa, los resultados de la instancia Large, puesto que los tiempos son relativamente buenos y no tiene problemas de algún resultado que no se tiene, por error en la memoria y similares.

Primero, observemos la Figura 1. Esta figura muestra la convergencia al óptimo, que sería 1, durante las 10 primeras iteraciones. Hay que aclarar primero, que la aleatoriedad a la hora de generar la población inicial juega un papel importante pero, aparte de esto, se pueden destacar varios puntos.

- La configuración 2 converge más rápidamente. Esto obedece a que para problemas no demasiado complejos, el reemplazo de los peores se comporta mejor que el reemplazo generacional.
- Las configuraciones 1 y 3 son iguales, excepto en el tamaño de la población y número generaciones. Esto no influye, en la convergencia en las 10 primeras iteraciones y queda reflejado en la figura, que convergen de manera muy similar.
- Otro punto característico de estas dos configuraciones es que la línea de convergencia hay un punto en el que baja en vez de ir en aumento en busca del óptimo. El porqué es muy sencillo, la selección probabilística hace que a veces el mejor sea descartado, que es lo que ha ocurrido en estos dos casos.

Configuración	Local	Micro	Small	Medium	Large	ExtraLarge	High CPU EL
Config. 1	99	583	220	111	134	108	106
Config. 2	664	4043	1280	715	797	634	669
Config. 3	1648	Error	3029	1667	1936	1527	1578

Cuadro 9. Tiempo de Ejecución en segundos. OneMax



Figura 1. Marea. Convergencia en instancia Large para las 3 configuraciones.

Sin nada más que referirnos, pasemos a observar una figura similar, pero correspondiente a la función Rastrigin. En este caso, no hace falta señalar que la convergencia hacia abajo se debe, a que es una función de minimización. En esta figura hay que destacar:

- La configuración 3 es la que se comporta mejor, de manera destacada en las 100 primeras iteraciones de Rastrigin. Por detrás de ella iría la configuración 2 y finalmente la uno parece que es la peor.
- En este caso podemos ver lo anteriormente comentado, parece que la mayor convergencia es fruto del azar y la probabilidad, puesto que las configuraciones 1 y 3 son la misma, pero es bastante probable, que al generar 50 mil individuos saldrá unos mejores individuos que al generar 10 mil como ocurre en la configuración número 1.

Por último, la Figura 3, muestra la convergencia durante 300 iteraciones de la función OneMax, que cuenta los unos de un cromosoma binario, hasta maximizar este número. El óptimo sería 512, que es el tamaño de la cadena elegido para observar los resultados.

- El mejor comportamiento es la configuración 3. Parece que aquí influye de nuevo el número de individuos de la población. Llega al óptimo casi unas 20 iteraciones antes que sus competidores.
- El reemplazo generacional de la configuración 2 vemos que en un inicio de algoritmo puede no ser muy bueno si no hay buena calidad en la población inicial, sin embargo observamos que la convergencia es constante, no como en el caso de la primera, que tras un buen inicio, converge más lentamente hasta ser la última en alcanzar el óptimo. Quizás el hecho de la menor presión de selección haya influido en esto.



Figura 2. Rastrigin. Convergencia en instancia Large para las 3 configuraciones.

7. Análisis de los resultados

Una vez vistos los resultados, analizando las tablas de tiempos, ficheros de resultados y gráficas obtenidas podemos sacar las siguientes conclusiones:

- **Limitación de Algunas instancias.** Algunas de las instancias, como era evidente no tienen memoria suficiente para realizar tareas complejas de cálculos. En nuestro caso, ha quedado demostrado en las instancias micro que no ha podido resolver Rastrigin con la configuración 2 y 3, y la small que no ha podido resolver la configuración 3 del mismo problema, ambas por falta de memoria, debido a que el problema Rastrigin trabaja con un gran número de números reales de precisión doble lo que hace ocupar gran parte de la memoria disponible.
- **Beneficioso.** En los demás casos, la utilización del cloud computing ha sido favorable debido a que los tiempos que nos han dado han sido muy parecidos, en muchos casos mejores, a los de nuestro pc, lo que lleva a pensar que para casos más complejos se podrían comportar mejor.
- **Trabajo Paralelo en Instancias.** Otra ventaja ha sido, que, en el mismo tiempo que se han realizado las ejecuciones para nuestro entorno local, han sido llevadas a cabo todas las ejecuciones en las seis instancias en la nube, ya que, las instancias han trabajado paralelamente y, lógicamente, sin influir unas con otras.
- **Selección.** Para el caso de la selección, en nuestro caso se ha comportado mejor la selección determinística por torneo, puesto que los problemas que hemos tratado no son muy complejos, se ha llegado al óptimo relativamente rápido y no han requerido de una amplia búsqueda ni en la salida de grandes óptimos locales, que son en los casos en los que la probabilística

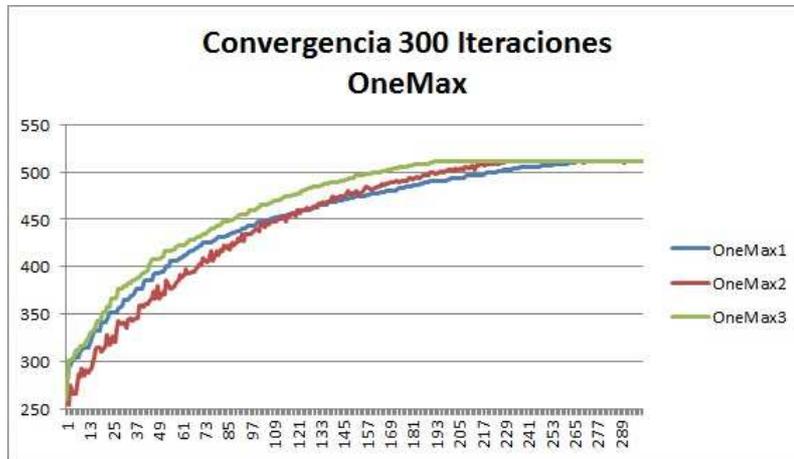


Figura 3. Rastrigin. Convergencia en instancia Large para las 3 configuraciones.

podría resultar favorable. Podemos apreciar una cosa curiosa en la selección probabilística, y es que cuando se ha alcanzado el óptimo, y en todas las iteraciones sea este el mejor individuo, hay alguna iteración en los que se genera uno peor a él y es seleccionado como mejor, quedando una gráfica y tanto rara.

- **Reemplazamiento.** El reemplazamiento generacional depende mucho de las tasas de selección, cruce y mutación elegidas. Si éstas son bajas, la población puede ir incluso decreciendo, pueso que la generación anterior es entera borrada y la nueva contendrá menos individuos. Esto hace que su comportamiento sea ligeramente más rápido en estos casos. Sin embargo, en nuestro problema ha resultado mejor el reemplazamiento de los nuevos individuos generados por los menos adaptados de la generación anterior (Worst), quizás por la misma razón que antes en la selección, no son problemas suficientemente complejos que requieran de una diversidad grande.
- **Cruce.** No se ha encontrado ningún resultado digno de mencionar en cuanto al cruce de uno o dos puntos. Por cuestiones puramente teóricas, podemos decir que el de dos puntos puede dar más diversidad a la población, pero no se ha visto que esto quede plasmado en nuestro estudio.
- **Mejorable.** Para muchos exigentes, esto no es suficiente, necesitan mejoras de tiempos evidentes y por ello se ha propuesto una alternativa mejor con Map Reduce.

8. Conclusiones y trabajo futuro

El Cloud Computing ha llegado con mucha fuerza y todo hace indicar que viene para quedarse. Hemos visto durante, la primera parte que son muchas más

las ventajas que nos ofrece este modelo que los inconvenientes a los que nos podemos enfrentar.

Todo proveedor de computación en la nube, nos ofrece una infraestructura que, a no ser que estemos en una gran empresa o seamos poseedores de superordenadores, no está a nuestro alcance y ello nos hace que instantáneamente podamos acceder a esos recursos y hacer uso de ellos de una forma rápida, sencilla y con un coste bajo. Precisamente esta posibilidad hace que sea una tecnología verdaderamente interesante para el mundo científico e investigador, pues habitualmente en tareas como análisis financiero, meteorológico o bioinformática, entre otras muchas cosas, se trabaja con ingentes cantidades de datos que requieren de millones de cálculos en poco tiempo y precisan de mucho tiempo, pese a tener grandes centros de datos.

Por ello en este proyecto se ha propuesto un pequeño estudio dentro del gran mundo que es la bioinformática, utilizando la infraestructura de Amazon, el mayor proveedor de Cloud Computing y mucho más configurable que Google App Engine, para la ejecución y análisis de algoritmos genéticos básicos.

Se ha implementado una pequeña librería para el diseño de algoritmos genéticos de forma secuencial y un completo manual para la ejecución de estos algoritmos en los servicios web de Amazon. Las características más importantes de esta propuesta han sido:

- **Extensibilidad:** Desde el primer momento de desarrollo se ha pretendido dotar a la librería de un carácter extensible, abstrayendo todo lo posible los distintos elementos dentro de un AG con el uso de interfaces, herencia y demás ventajas de un lenguaje como Java y facilitando el prototipado rápido de AGs mediante la implementación de un número mínimo de funciones.
- **Portabilidad:** La aplicación se encapsula en un fichero JAR que puede ser ejecutado en cualquier entorno anfitrión que disponga de un compilador Java, independientemente de su sistema operativo. Esto se ha hecho así, porque el objetivo principal era ver las ventajas de computación en la nube, y lo mejor era un sistema portable entre distintas instancias volátiles en Amazon EC2.
- **Sencillez:** El objetivo principal del trabajo, era conocer un poco más distintas posibilidades de servicios de la nube de Amazon y como facilita el análisis de datos y no profundizar mucho en Algoritmos Genéticos, por esta razón se han implementado tres funciones sencillas. El único conocimiento necesario para realizar las pruebas viene explicado detalladamente en el documento adjunto o la documentación generada por cada proyecto.
- **Aprendizaje:** Este trabajo es válido tanto como para alguien quiera iniciarse en el conocimiento de los algoritmos genéticos, como para alguien que quiere sentar las bases de Cloud Computing tener acceso a instancias virtuales en Amazon.

Una vez vistos hecho el análisis de los resultados de la versión secuencial surgió la posibilidad de buscar una alternativa paralela para un algoritmo genético e investigando se encontró la posibilidad de usar MapReduce, por ello, se ha des-

critico teóricamente en que consiste este modelo y en concreto, la implementación por parte de Hadoop, que es la más conocida.

Para introducirnos en el mundo de Hadoop, se ha explicado como hacer una configuración pseudo-distribuida en nuestro pc para comenzar a escribir aplicaciones de este tipo. Como primer ejemplo de ello, se ha propuesto y explicado la implementación de un básico contador de frecuencias de palabras en un conjunto de archivos, que pese a ser sencillo, nos da una muestra del potencial de MapReduce y de la sencillez de su implementación.

Como extensión se ha presentado una aproximación a un sistema capaz de adaptar la naturaleza intensiva en datos del modelo MapReduce con el carácter iterativo de un Algoritmo Genético, sentando las bases para construir un software que permita ejecutar de forma distribuida algoritmos genéticos paralelos tanto sobre clústers, como sobre el servicio que nos proporciona Amazon Elastic MapReduce.

En líneas generales, se ha descrito el diseño y la implementación de AGs básicos tanto de forma secuencial como paralela utilizando entornos secuenciales y de cloud computing con Amazon como proveedor, y se ha introducido al uso del Framework de Hadoop para el diseño de aplicaciones con MapReduce.

Por último, destacar que las pruebas realizadas con la batería de problemas elegida (“Rastrigin”, “OneMAX” y “Marea”) arrojan resultados satisfactorios en cuanto a escalabilidad y tiempo, teniendo en cuenta además que nuestra computadora desde la que mandamos los trabajos no consume recurso alguno.

Futuras líneas de trabajo

Hasta el momento solo ha podido desarrollarse un modelo de cómo sería la implementación de un algoritmo genético paralelo usando MapReduce. La principal línea de trabajo sería terminar la implementación de este modelo y hacer extensible las clases implementadas para la parte secuencial a la parte paralela.

Dentro de la línea seguida en el proyecto podemos contemplar otras alternativas que completarían el estudio del modelo de programación MapReduce:

- Se podría investigar si compensa la generación inicial de individuos con una tarea MapReduce como propone [10]. Esto sería útil para utilizar poblaciones de gran tamaño ya que en este caso, tendría un costo inicial muy alto para el nodo maestro mientras los otros se encontrarían ociosos.
- Otro aspecto que podría ser estudiado es la definición de una clase personalizada InputFormat (la encargada de describir la especificación de los datos de entrada de un trabajo MapReduce) para poder configurar el número de tareas map a gusto del usuario.

En lo que engloba al Cloud Computing, hay varias opciones que nos gustaría llevar a cabo:

- Mover una aplicación a la nube, migrar sus datos, configurar una instancia para que no sea volátil, contratando una ip estática y que ésta actúe como servidor, y montar una página web en ella.
- Utilizar proveedores PaaS, como Windows Azure o Google App Engine para desarrollar un servicio en la web.

Referencias

1. Sosinsky, B.: Cloud Computing Bible. First edn. John Wiley & Sons Ltd (2011)
2. Miller, M.: Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. First edn. Que (2008)
3. Formigacloud: Evaluación de amazon ec2. (2011)
4. J. Dean, S.G.: Mapreduce: Simplified data processing on large clusters. (2008)
5. Bagley, J.: The Behavior of adaptive systems which employ genetic and correlation algorithms. PhD thesis, University of Michigan (1967)
6. Holland, J.: Adaptation in Natural and Artificial Systems. Michigan Press (1975)
7. O. Cerdón, L. Magdalena, F.H.: Ten years of genetic fuzzy systems: current framework and new trends. (2003)
8. Mitchell, M.: An introduction to genetic algorithms. (1998)
9. Jong, K.A.D.: An analysis of the behaviour of a class of genetic adaptive systems. PhD thesis, PhD Tesis, University of Michigan (1975)
10. A. Verma, X. Llorca, D.G.: Scaling genetic algorithms using mapreduce. (2009)

Estrategia de Aprendizaje Activo y Cooperativo para Periféricos y Dispositivos de Interfaz Humana en el Grado de Ingeniería Informática

Fernando Rojas

Departamento de Arquitectura y Tecnología de Computadores. ETSI Informática y de Telecomunicación. Universidad de Granada.
frojas@ugr.es

Resumen. Esta contribución presenta una propuesta para la docencia y aprendizaje de la asignatura “Periféricos y Dispositivos de Interfaz Humana” correspondiente al Grado en Ingeniería Informática de la Universidad de Granada. La metodología docente planteada se basa en el aprendizaje cooperativo y activo, de forma que el estudiante pueda adquirir las competencias establecidas para esta asignatura mediante la experimentación con casos prácticos colaborando en proyectos comunes con sus compañeros. Esta estrategia de aprendizaje permite una dependencia positiva entre los estudiantes y una mejora en el proceso de aprendizaje al experimentar con casos prácticos reales.

Palabras Clave: Espacio Europeo de Educación Superior (EEES), Ingeniería Informática, sistemas de información, periféricos, aprendizaje cooperativo, aprendizaje colaborativo, aprendizaje activo.

Abstract. This paper shows a collaborative and active approach for the teaching and learning of “Computer Peripherals and Human Interface Devices” for the Degree in Computer Science from the University of Granada. The proposed teaching methodology is based on active and cooperative learning, so that students can acquire the skills set for this subject by experimenting with case studies on common projects collaborating with peers. This methodology allows for a positive dependence between students and an improvement in the learning process by experimenting with real applications.

Keywords: European Higher Education Area (EHEA), Computer Science, information systems, computer peripherals, cooperative learning, collaborative learning, active learning.

1 Introducción

1.1 Objetivos de aprendizaje

Uno de los principales objetivos de la universidad es la formación de un alumno interactivo, reflexivo, crítico, que construya su propio conocimiento con el apoyo del profesor y de sus compañeros, profesionalmente competente y capaz de potenciar su esfuerzo mediante el trabajo en equipo [1]. Para alcanzar este objetivo se precisa una cuidadosa planificación que integre las variables relevantes del proceso de enseñanza-aprendizaje y que suministre a profesor y alumnos la información necesaria acerca de lo que se trata de conseguir, cómo lograrlo y de qué modo se van a verificar los resultados, acrecentando con ello la motivación de los alumnos y reforzando el papel facilitador del profesor [2].

El término aprendizaje colaborativo o cooperativo se refiere al sistema de aprendizaje basado en metodologías de trabajo en grupo que impulsan al grupo a cooperar hacia el logro de un mismo objetivo [3]. En un escenario de aprendizaje colaborativo cada uno de los copartícipes está comprometido en la búsqueda de soluciones parciales en busca de un objetivo común y su contribución al grupo no es competitiva sino que genera una interdependencia positiva. De esta forma, el logro de un resultado es más importante que las contribuciones individuales de cada sujeto [4].

En un modelo presencial como el que se propone para la docencia y aprendizaje de la asignatura “Periféricos y Dispositivos de Interfaz Humana” en el Grado de Ingeniería Informática la aplicación de una metodología de aprendizaje activo y colaborativo debe redundar en un mejor desarrollo de las competencias atribuidas a esta asignatura por parte del estudiante.

La competencias específicas que se fijan para esta asignatura son [5, 6]:

- Capacidad de integrar soluciones de Tecnologías de la Información y las Comunicaciones y procesos empresariales para satisfacer las necesidades de información de las organizaciones, permitiéndoles alcanzar sus objetivos de forma efectiva y eficiente, dándoles así ventajas competitivas.
- Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

1.2 Ubicación de la asignatura en el plan de estudios

La asignatura “Periféricos y Dispositivos de Interfaz Humana” se enmarca como asignatura optativa dentro del plan de estudios del Grado en Ingeniería Informática que se imparte en la Universidad de Granada en el módulo “Complementos de Sistemas de Información” (Figura 1).

ASIGNATURA	MÓDULO	MATERIA	CURSO	SEMESTRE	CRÉDITOS	TIPO
Periféricos y Dispositivos de Interfaz Humana	Complementos de Sistemas de Información	Complementos Tecnológicos y de Infraestructuras en Sistemas de Información	4º	8º	6	Optativa
Proyecto Fin de Grado						12
Formación Complementaria Interdisciplinar			Prácticas en Empresa			30
Complementos Especialidad 1	Complementos Especialidad 2	Complementos Especialidad 3	Complementos Especialidad 4	Complementos Especialidad 5		
Especialidad 1: Computación y Sistemas Inteligentes	Especialidad 2: Ingeniería del Software	Especialidad 3: Ingeniería de Computadores	Especialidad 4: Sistemas de Información	Especialidad 5: Tecnologías de la Información	48	
Obligatorias de rama						90
Formación básica						60

Figura 1. Estructura de la Especialidad Sistemas de Información dentro del Grado en Ingeniería Informática [5].

Si bien esta asignatura pertenece a la especialidad “Sistemas de Información”, su carácter optativo y multidisciplinar la hace perfectamente plausible para ser elegida como parte de la Formación Complementaria Interdisciplinar de un estudiante que provenga de cualquiera de las otras cuatro especialidades.

La oferta de optatividad en el módulo “Complementos de Sistemas de Información” incluye asimismo las siguientes asignaturas: “Sistemas de Información Geográficos”, “Gestión de Recursos Digitales”, “Sistemas cooperativos y Gestión de Contenidos” y “Redes y sistemas complejos” (ver Figura 2, derecha).



Figura 2. Oferta de asignaturas obligatorias y optativas para la Especialidad Sistemas de Información dentro del Grado en Ingeniería Informática.

El programa oficial de teoría y prácticas de la asignatura “Periféricos y Dispositivos de Interfaz Humana” del plan de estudios del Grado en Ingeniería Informática se puede encontrar en la web de grados de la Universidad de Granada [6] y se resume en la Tabla 1.

Tabla 1. Programa de teoría y prácticas de la asignatura “Periféricos y Dispositivos de Interfaz Humana”

Temario Teórico	Temario Práctico
Periféricos para almacenamiento masivo.	Introducción
Transductores de entrada.	Programación de rutinas de E/S
Dispositivos de entrada convencionales.	Determinación de características de un transductor de entrada.
Dispositivos de salida convencionales.	Experimentación con un sistema <i>Brain Computer Interface</i> (BCI)
Periféricos para sistemas multimedia.	
Periféricos para realidad virtual.	
Periféricos para interacción persona-computador en nuevos entornos.	

2 Periféricos: motivación al estudiante

Los periféricos constituyen algunos de los elementos esenciales en un sistema informático que, sin embargo, a menudo son obviados frente a otros elementos tales como placas madre, memorias y microprocesadores. No obstante, los periféricos tienen un gran impacto sobre el rendimiento global del sistema. De hecho, como se afirma en [7] *“un computador sin dispositivos de E/S es como un automóvil sin ruedas --no se puede ir muy lejos sin ellas. Y aunque el tiempo de CPU es interesante, el tiempo de respuesta es seguramente una mejor medida del rendimiento. El cliente que paga un computador se preocupa por el tiempo de respuesta aunque el diseñador de CPU no lo haga”*.

Tradicionalmente se ha medido la “inteligencia” de un computador como la capacidad intensiva de cálculo que éste posee. En este aspecto, sin duda, el computador es superior al ser humano. Por ejemplo, realizar la raíz cuadrada de números enteros de 20 dígitos es una tarea trivial para un computador. No lo es tanto para un ser humano.

Sin embargo, la habilidad lógica y matemática es tan sólo una de las (al menos) ocho dimensiones de la inteligencia humana (Teoría de las Inteligencias Múltiples, Howard Gardner [8], Figura 3). Sin embargo, en el resto de dimensiones, los computadores se encuentran aún muy lejos de la capacidad humana. La tarea que tratan de llevar a cabo los periféricos de un computador es mejorar alguno de los aspectos de estas “otras dimensiones” (visual, lingüística, emocional, etc.).

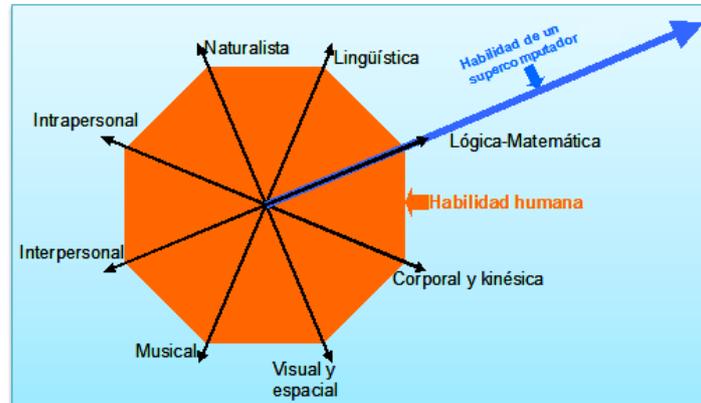


Figura 3. Teoría de las inteligencias múltiples (H. Gardner [8]) y estado de los computadores actuales respecto a la inteligencia humana.

En esta asignatura se estudian los periféricos de computador atendiendo a varios puntos de vista: físico, lógico y funcional. Se analizarán algunos de los periféricos más usados (discos duros, dispositivos de entrada, monitores, etc.). A través del proyecto docente propuesto, se trata de situar al estudiante en contacto directo con el funcionamiento interno de un periférico, experimentando de forma activa con diferentes dispositivos de conexión a un computador.

Se pretende presentar proyectos atractivos de programación y uso para los estudiantes que puedan llevar a cabo mediante dispositivos relativamente fáciles de aprender a manejar y diseñados expresamente para su utilización con fines docentes y de autoaprendizaje.

3 Aprendizaje cooperativo e interacción con periféricos reales

En la asignatura se tratará de proporcionar una visión actual y aplicada de los periféricos disponibles en la actualidad. La asignatura se divide a partes iguales entre clases teóricas-expositivas (2 horas semanales) y clases prácticas de laboratorio (2 horas semanales). Sin renunciar de ninguna manera a una metodología interactiva en las clases teórico-expositivas, es en las clases prácticas en las que se podrá desarrollar con mayor potencial la interacción y el aprendizaje mediante el uso de periféricos reales.

En este sentido, se propone el uso de la plataforma de electrónica abierta “Arduino” (<http://arduino.cc/>) [9] y las placas de interfaz USB de experimentación “Velleman” (<http://www.velleman.eu>) [10].

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Arduino puede adquirir información del entorno mediante diferentes sensores a través de sus conexiones de entrada (periférico de entrada) y puede interactuar con su entorno que le rodea controlando luces, motores y otros actuadores (periférico de salida). El

microcontrolador en la placa Arduino se programa mediante un lenguaje de programación y un entorno de desarrollo propio de fácil aprendizaje. Debido a su naturaleza como plataforma de código abierto y hardware libre, así como de su aplicabilidad inmediata para proyectos sencillos, resulta un apoyo ideal para la docencia práctica en la asignatura “Periféricos y Dispositivos de Interfaz Humana”.



Figura 4. Microcontrolador Arduino Uno (Imagen obtenida de <http://http://arduino.cc>)

Las tarjetas de interfaz USB de experimentación “Velleman” (Figura 5.) por su parte complementan a las anteriores “Arduino” permitiendo un estilo de programación de más alto nivel, permitiendo crear aplicaciones gráficas que realizan pequeñas interacciones con el entorno como control de motores, relés, sensores de temperatura, presión, etc. Además, de esta forma los estudiantes experimentan y contrastan con el uso de entornos diferentes para la programación y manejo de microcontroladores, obteniendo una visión mucho más amplia y motivadora del uso de los dispositivos periféricos que la que obtendrían con el uso exclusivo de los periféricos clásicos de un computador (teclado, ratón, pantalla, etc.).



Figura 5. Tarjeta de interfaz USB de experimentación “Velleman” VM110N (Imagen obtenida de <http://www.velleman.eu>)

La utilización de este material se encuentra pendiente de la correspondiente adjudicación del presupuesto solicitado a través del XXI Programa de Apoyo a la Docencia Práctica de la Universidad de Granada. La utilización de este material permite una interacción directa del estudiante con los dispositivos periféricos de un computador. La motivación e involucración del estudiante se encuentran prácticamente aseguradas mediante la realización de pequeños proyectos muy

atractivos como el control remoto de un vehículo de juguete, el accionamiento programado de luces, el control de temperatura mediante sensores, etc.

Se espera que el equipamiento mediante este material para la asignatura mejore la experiencia de los estudiantes y la conexión entre los conceptos adquiridos de manera más teórica con la aplicación real en diferentes sistemas de computadores.

4 Evaluación

Aunque los porcentajes exactos de ponderación de cada parte de la asignatura se especificarán de forma exacta al inicio del curso 2013-2014, se primará especialmente el trabajo aplicado en proyectos reales y cooperativos.

Para la parte teórica se realizarán exámenes finales o parciales, sesiones de evaluación y entregas de ejercicios sobre el desarrollo y los resultados de las actividades propuestas. Para la parte práctica se realizarán prácticas de laboratorio, resolución de problemas y desarrollo de proyectos (preferentemente en grupo), y se valorarán las entregas de los informes/memorias realizados por los alumnos, o en su caso las entrevistas personales con los alumnos y las sesiones de evaluación.

La realización de proyectos de prácticas que demuestren inequívocamente la superación de conceptos de la parte teórica pueden incluso aumentar la ponderación de la parte práctica, puesto que con esta se está asimilando de forma aplicada y mucho más consolidada la parte teórica (*“Lo que oigo, olvido. Lo que veo, recuerdo. Lo que hago, aprendo”*, Confucio).

5 Conclusiones

La metodología docente propuesta para la asignatura de nueva impartición en el cuarto curso del Grado en Ingeniería Informática de la Universidad de Granada “Periféricos y Dispositivos de Interfaz Humana” pretende enfocar el aprendizaje del estudiante a través de casos prácticos en los que este participe activamente y en colaboración con sus compañeros. En este sentido, el aprendizaje activo y cooperativo se presentan como las estrategias de aprendizaje idóneas para esta asignatura, sobre todo en su parte práctica.

La metodología propuesta permite una dependencia positiva entre los estudiantes y una mejora en el proceso de aprendizaje al experimentar con proyectos tangibles y de aplicabilidad en el mundo real. El objetivo de la asignatura es que el alumno comprenda la importancia de los dispositivos periféricos de un computador frente a la tradicional visión de un computador basada en el rendimiento de su unidad central de proceso.

Referencias

1. Camacho, S.: Formación del profesorado y nuevas tecnologías. In: Sáenz Barrio, Ó., Rodríguez Dieguez, J.L. (eds.) Tecnología educativa: nuevas tecnologías aplicadas a la educación, pp. 413-444. Editorial Marfil, Alcoy (1995)
2. Libro Blanco del Título de Grado en Ingeniería Informática, www.aneca.es/Documentos-y-publicaciones/Libros-Blancos
3. Ovejero Bernal, A.: El aprendizaje cooperativo : alternativa eficaz a la enseñanza tradicional. PPU, Barcelona (1990)
4. Scagnoli, N.I.: Estrategias para Motivar el Aprendizaje Colaborativo en Cursos a Distancia. Illinois Research and Scholarship (Open Community) <http://hdl.handle.net/2142/10681>, (2005)
5. Universidad de Granada, Programa Verifica, Grado en Ingeniería Informática, <http://grados.ugr.es/informatica/pages/infoacademica/archivos/verificaingenieriainformatica>
6. Universidad de Granada, Guía Docente de Periféricos y Dispositivos de Interfaz Humana, Grado en Ingeniería Informática, http://grados.ugr.es/informatica/pages/infoacademica/guias_docentes/optespsi/perifericosdispointerfazhumana
7. Hennessy, J.L., Patterson, D.A.: Arquitectura de computadores: un enfoque cuantitativo. McGraw-Hill, Madrid, [etc] (2002)
8. Gardner, H.: Frames of mind : the theory of multiple intelligences. Basic Books, New York (1983)
9. Plataforma Arduino, <http://www.arduino.cc/>
10. Tarjetas de interfaz USB de experimentación <http://www.velleman.eu/products/view/?id=351980>

Propuesta docente para la nueva asignatura de Tecnologías Emergentes en el Grado en Ingeniería Informática

Samuel Romero¹

¹ Departamento de Arquitectura y Tecnología de Computadores. ETSI Informática y de Telecomunicación. Universidad de Granada
sromero@ugr.es

Resumen. En este artículo se expone el diseño de la propuesta docente para la nueva asignatura de “Tecnologías Emergentes” que empezará a impartirse en el seno de la especialidad en Ingeniería de Computadores como optativa, en el curso 2013-14. La asignatura, sin precedentes como tal en otras titulaciones similares de nuestro país, tratará de transmitir al alumnado la importancia de permanecer actualizados y familiarizados con las tecnologías en fase emergente. Se enseñará a analizar dichas tecnologías en cuanto a expectativas, viabilidad, y aplicaciones. A modo de ejemplo, se aproximará a los estudiantes a tres tecnologías emergentes: Sistemas vestibles, redes de sensores inalámbricos, e interfaces cerebro-computador. El enfoque global de la asignatura es eminentemente práctico y participativo, buscando dotar al estudiante de las habilidades necesarias para el desempeño de su profesión.

Palabras Clave: tecnologías emergentes, sistemas vestibles, redes de sensores inalámbricos, interfaces cerebro-máquina, BCI, Arduino, metodología docente.

Abstract. In this paper, we depict the design for the teaching proposal for a new course on “Emerging Technologies”, to start within the minor on Computer Engineering, in the academic year 2013-14. This subject, without precedents as this one in similar degrees in our country, aims to make the students perceive the relevance of keeping updated and familiarized with respect to technologies in emerging phase. We will train the students in the analysis of such technologies regarding expectancies, viability and applications of them. As examples, we will introduce the students into three emerging technologies: wearable systems, wireless sensors networks, and brain-computer interfaces. The global focusing for the course is mainly practical and interactive, seeking to provide the students with the skills that are required to perform adequately in their professional life.

Keywords: emerging technologies, wearable systems, wireless sensors networks, brain-computer interfaces, BCIs, Arduino, teaching methodology.

1 Introducción

En este artículo se presenta una propuesta didáctica para una nueva asignatura del Grado en Ingeniería Informática, denominada “Tecnologías Emergentes”. Esta asignatura se impartirá por primera vez en el próximo curso 2013-2014. En los siguientes apartados se realiza una descripción de las características de la asignatura, según se recoge en el plan de estudios. Se realiza una breve panorámica de los contenidos que componen la materia, y su interrelación con otras asignaturas de la carrera y de la especialidad en Ingeniería de Computadores.

Además de contenidos propiamente relacionados con determinadas tecnologías emergentes, y que, por su propia naturaleza, están sujetas a evolución o incluso extinción, se pretende dotar al estudiante de las habilidades necesarias para localizar y detectar tecnologías en fase emergente, analizar su viabilidad y aplicaciones, y ser capaces de familiarizarse rápidamente con éstas.

Se describen, conforme al diseño de la ficha VERIFICA, las competencias y objetivos que se han de cubrir con esta asignatura, y se detallan los contenidos teóricos y prácticos que se han previsto para la misma.

Finalmente, se expone la metodología docente que se pretende emplear, con enfoque novedoso y participativo. Se describe el sistema de evaluación previsto para esta asignatura y se comenta brevemente la bibliografía que se va a emplear.

2 La nueva asignatura de Tecnologías Emergentes

2.1 Contexto

Con la llegada de los nuevos planes de estudios que se derivan del proceso de Bolonia, en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicaciones de la Universidad de Granada se ha venido implantando en los últimos años un nuevo título de Grado en Ingeniería Informática.

Este título cuenta con varias intensificaciones o especialidades de rama, entre las que se encuentra la de Ingeniería de Computadores, que mayoritariamente se imparte desde el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada.

En el cuarto y último curso de esta titulación, y dentro de la materia denominada “Complementos de Sistemas de Cómputo para Aplicaciones Específicas”, se oferta la materia “Tecnologías Emergentes” [1], con carácter optativo.

Esta asignatura tiene previsto su comienzo en el primer semestre del curso académico 2013-2014. Si bien está adscrita al Grado en Ingeniería Informática, se puede ofertar también en el Grado en Ingeniería de Tecnologías de Telecomunicación.

Para cursar esta asignatura no se han establecido prerequisites obligatorios, si bien sería beneficioso para aquellos estudiantes que la vayan a cursar el haber estudiado antes otras materias como Sistemas con Microprocesador y Sistemas Embebidos, entre otras.

2.2 Contenidos principales de la asignatura

Atendiendo a lo recogido en la ficha VERIFICA, los contenidos básicos de la asignatura son los siguientes:

- Hardware para sistemas vestibles.
- Redes de sensores inalámbricos.
- Sistemas para rehabilitación sensorial.
- Sistemas implantables.
- Aplicaciones biomédicas.
- Interfaces cerebro-máquina.
- Conformidad y certificación de la tecnología

Como se puede ver, los contenidos de Tecnologías Emergentes giran, principalmente, en torno a tres tecnologías modernas:

- **Sistemas vestibles** (*Wearable Computing* o *e-Textiles*)
- **Redes de Sensores Inalámbricos** (*Wireless Sensor Networks*, IoT, M2M)
- **Interfaces Cerebro-Máquina** (BCI o BMI, *Neuroengineering*)

Los sistemas vestibles [2] se refieren a aquellos elementos electrónicos (computador, microcontrolador, sensores, actuadores, etc.) que se pueden incorporar en la vestimenta del usuario, con muy diversos fines. Así, podemos encontrar ejemplos de estos sistemas en ropa “inteligente” que monitorice la salud de un paciente, sistemas de señalización para vestimenta nocturna en carreteras (ciclistas, agentes de tráfico, operarios de mantenimiento), calzado deportivo inteligente que registre nuestra actividad, o complementos para personal militar que faciliten la comunicación con otros y la localización de objetivos.

Todos estos sistemas presentan una serie de requisitos comunes en cuanto al tipo de sistemas que deben incorporar: bajo consumo, fácil integración en elementos textiles, escaso volumen, interfaces integrables y cómodas, etc.

Por otra parte, las redes de sensores inalámbricos [3] se relacionan con tecnologías en las que elementos de muy bajo consumo, empujables en distintos dispositivos y con capacidad para hacer interfaz con sensores analógicos o digitales, incluyen la capacidad de crear redes de estos nodos de sensores que se pueden intercomunicar siguiendo diferentes estándares. En esta línea se han generado nuevos términos como “Internet of Things” (IoT) y comunicación M2M (“Machine-to-machine”).

Las aplicaciones de estas redes de sensores inalámbricos pueden ir desde la monitorización ambiental de áreas extensas (ciudades, plantas industriales, zonas forestales), domótica, comunicación entre vehículos, entre robots autónomos, etc.

Por último, se contempla el estudio de principios de Neuroingeniería [4] para el desarrollo de sistemas que permitan comunicar el cerebro o el sistema nervioso con un computador (Interfaces cerebro-máquina o cerebro-computador). De esta forma, mediante una serie de electrodos, se puede registrar la actividad mental y usarla para controlar distintos elementos como sillas de rueda, sistemas domóticos, interfaces de comunicación, etc.

2.3 Un enfoque con vista al futuro

Si bien los contenidos principales de la asignatura se centran en las tres tecnologías anteriormente mencionadas, el propio carácter de la asignatura se presta a una constante evolución, con la extinción de tecnologías ya superadas y el nacimiento de otras nuevas.

Por tanto, será necesario, por una parte, el actualizar este catálogo de tecnologías emergentes con las que los estudiantes se deben familiarizar. Pero por otra parte, se busca dotar a estos estudiantes de habilidades no tan efímeras como puedan serlo algunas de estas tecnologías.

Nos referimos a la capacidad en sí de detectar nuevas tecnologías en fase emergente, analizar su viabilidad y posibilidades de supervivencia, su aplicabilidad, y en definitiva, el desarrollo del pensamiento crítico necesario en el ámbito de la ciencia y la tecnología, y que resulta imprescindible para discernir el verdadero interés y potencial de las tecnologías según se presentan en diferentes fuentes de información.

De esta forma, buscamos que el estudiante analice tecnologías en el pasado han fracasado a pesar de su potencial (el video Beta, el miniDisc, por ejemplo), el tiempo de vida cada vez más corto de las tecnologías (por ejemplo, discos duros que se van reemplazando por estado sólido), y la fiabilidad de las fuentes de información sobre nuevas tecnologías (por ejemplo, medios de comunicación de masas vs. Revistas especializadas como el MIT Technology Review [5]).

3 Diseño formal de la asignatura

3.1 Competencias

Según se recoge en la ficha VERIFICA, en la asignatura de Tecnologías Emergentes se han seleccionado las siguientes competencias para su desarrollo en el alumnado:

- **Competencias Específicas de la Asignatura**

IC1. Capacidad de diseñar y **construir sistemas** digitales, incluyendo computadores, sistemas **basados en microprocesador y sistemas de comunicaciones**.

IC5. Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuadas para el soporte de aplicaciones empotradas y de tiempo real.

IC6. Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.

IC7. Capacidad para analizar, evaluar, seleccionar y configurar plataformas hardware para el desarrollo y ejecución de aplicaciones y servicios informáticos

- **Competencias Específicas del Título**

E1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

E4. Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.

E7. Capacidad para conocer, comprender y aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico en Informática y manejar especificaciones, reglamentos y normas de obligado cumplimiento

E8. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

3.2 Objetivos

Se pretende que los estudiantes que hayan cursado con éxito la asignatura de Tecnologías Emergentes, sean capaces de:

- Analizar las nuevas tecnologías en Ingeniería de Computadores. Detectar tendencias emergentes en fase de investigación. Conocer las líneas de investigación en auge en Ingeniería de Computadores. Familiarizarse con estas novedades tecnológicas.
- Caracterizar un sistema vestible, identificar sus requisitos en función de la aplicación concreta y diseñar un sistema completo. Identificar ejemplos de sistemas vestibles en fase de investigación y desarrollo.
- Conocer, concebir y desplegar redes de sensores inalámbricos. Comprender las tecnologías de comunicación inalámbrica y los protocolos usados. Detectar aplicaciones en las que estas redes puedan ser de utilidad.
- Conocer principios básicos de ingeniería biomédica. Comprender el papel de la Ingeniería de Computadores en biomedicina.
- Familiarizarse con las técnicas de tratamiento de información biológica, en especial con señales del sistema nervioso.
- Clasificar las interfaces cerebro-máquina, conocer los sistemas de referencia para la obtención de señales cerebrales, identificar los distintos tipos de señales y su posible modulación mediante la voluntad, y definir el sistema de clasificación de señales y/o patrones cerebrales.
- Conocer la normativa y certificaciones que han de cumplir dispositivos como los que se estudian en la asignatura (sensores inalámbricos, implantes activos, dispositivos biomédicos).

3.3 Contenidos

El programa de contenidos concretos previstos para la asignatura, contempla los siguientes temas:

Tema 1. Sistemas Vestibles.

- 1.1. Computación Ubicua y Vestible
- 1.2. Aplicaciones de los sistemas vestibles.
- 1.3. Hardware para sistemas vestibles.

Tema 2. Redes de sensores.

- 2.1. Sensores y redes. Arquitectura de red.
- 2.2. Esquemas de programación.
- 2.3. Estándares y sistemas actuales
- 2.4. Aplicaciones de las redes de sensores.

Tema 3. Sistemas de Rehabilitación Sensorial.

- 3.1. Ingeniería de la Rehabilitación
- 3.2. Robótica Asistencial
- 3.3. Sustitución y apoyo sensorial

Tema 4. Sistemas Implantables. Aplicaciones Biomédicas

- 4.1. Principios de Ingeniería Biomédica.

- 4.2. Implantes y sistemas implantables.
- 4.3. Aplicaciones biomédicas implantables.

Tema 5. Interfaces Cerebro-Máquina.

- 5.1. Concepto y tipos.
- 5.2. ICM no invasivos.
- 5.3. ICM invasivos.
- 5.4 Ejemplos.

Tema 6. Conformidad y Certificación de la Tecnología

- 6.1 Regulación y certificación de redes de sensores
- 6.2 Organismos certificadores de sistemas implantables
- 6.3 Certificación de implantes y sistemas biomédicos.

4 Metodología docente

3.1 Clases teóricas

Las clases de teoría de la asignatura se organizarán en dos fases. En la primera fase, se combinarán las lecciones magistrales habituales para sentar los conceptos básicos con la puesta en común de trabajo autónomo del alumnado. Este trabajo autónomo estará previamente preparado a partir de una serie de guías (se incluye un ejemplo como anexo) que orientarán al estudiante en la aproximación al tema que se va a impartir, de manera que pueda averiguar por su cuenta conceptos, ejemplos y técnicas que luego podrá compartir y discutir con el resto de la clase. Esta técnica hará las clases teóricas más participativas, evitando el clásico monólogo del profesor, a la vez que permitirá potenciar la capacidad de aprendizaje autónomo del estudiante, tan necesaria en una profesión que requiere constante actualización y más en el curso previo a su incorporación al mercado laboral.

Como anexo a este artículo, se plantea un ejemplo de guía de trabajo autónomo, escrito en un lenguaje accesible, con dificultad progresiva y que pretende motivar al estudiante en su aproximación activa al próximo tema que se ha de ver en clase.

Además, en esta primera fase, se intercalarán una serie de seminarios:

- Seminario 1: análisis crítico de la información
- Seminario 2: fuentes de información sobre tecnologías emergentes
- Seminario 3: Arduino, fundamentos básicos
- Seminario 4: Android (en función de conocimientos iniciales)
- Seminario 5: cómo hacer una presentación (idem)

Una vez concluida esta primera fase, se dedica la última parte de la asignatura a una segunda fase en la que los estudiantes realizan presentaciones ante la clase acerca de una tecnología en fase emergente. Esta presentación versará sobre un tema previamente seleccionado y supervisado en una tutoría específica. Deberá incluir una aproximación a esta tecnología, así como un análisis de la misma (fundamentos, viabilidad, expectativas, aplicaciones, fuentes de información).

3.2 Clases prácticas

Las sesiones de prácticas de la asignatura se organizarán también en dos fases. En la primera fase, se realizarán una serie de prácticas-tutoriales basados en guiones, al objeto de que los estudiantes se familiaricen con el uso básico de estas tecnologías. Se proponen las siguientes prácticas guiadas:

- Práctica 1: Sistemas vestibles: hardware para dispositivos vestibles, basados en LillyPad Arduino [6]. Demostración de montaje y programación de un chaleco con señalización nocturna basado en microcontrolador.
- Práctica 2: Redes de sensores inalámbricos: montaje de un sistema basado en comunicación entre motas inalámbricas basado en XBee [7] y Arduino
- Práctica 3: Uso de plataformas BCI bajo coste: Neurosky Mindset [8] y Emotiv EPOC [9] (registro de señales de actividad cerebral, y programación para control de un dispositivo).

En la segunda fase, los estudiantes, por grupos cuya composición dependerá del número total de estudiantes, trabajarán en un pequeño proyecto libre basado en la aplicación de alguna de las tecnologías vistas en la primera fase o incluso una combinación de ellas. La selección de este proyecto será monitorizada en tutoría previa, y se hará un seguimiento de su desarrollo en las sesiones de prácticas. Al finalizar, los estudiantes presentarán a la clase sus resultados y conclusiones.

3.3 Método de evaluación

Para la evaluación de los estudiantes que sigan esta asignatura, se ha planteado el uso de los siguientes mecanismos y su ponderación en la calificación final:

- La nota final se compondrá de: Teoría (50%) + Prácticas (50%)
- Teoría
 - Participación en clase de teoría (resultado del trabajo autónomo) 10%
 - Tests elaborados colaborativamente (al final de cada tema) 20%
 - Exposición final sobre una tecnología (usando una rúbrica) 20%

- Prácticas
 - Asistencia a sesiones prácticas 20 %
 - Exposición del proyecto desarrollado (usando una rúbrica) 30%

De esta forma, se estimula la participación del estudiante en el desarrollo de la asignatura, y se le hace co-partícipe de la evaluación, dado que intervendrá proponiendo preguntas de test para su inclusión en una base de cuestiones de las que se alimentarán automáticamente los tests de cada tema, que se harán por ordenador. Este esquema ha sido ensayado anteriormente con muy buenos resultados [10].

Es muy conveniente el uso de “rúbricas” [11] o plantillas de criterios de evaluación, de los que dispondrán con antelación los estudiantes, de manera que sepan qué deben hacer para obtener una determinada calificación, y qué aspectos se deben cuidar para sus trabajos (por ejemplo, en la exposición del proyecto, atenerse al tiempo preestablecido para sacar la máxima calificación, o no leer las transparencias).

Además de la evaluación habitual a la que son sometidos los estudiantes, se incluirán mecanismos para evaluar la acción del docente y el diseño de la asignatura, de cara a proporcionar una retroalimentación que redunde en una mejor docencia. En condiciones ideales, se realizará una encuesta en un punto intermedio del semestre (de forma que las posibles mejoras puedan aplicarse en ese mismo curso) y otra al final (para perfilar el diseño de cara al curso siguiente).

3.4 Bibliografía

La bibliografía que se empleará en esta asignatura se ha seleccionado teniendo en cuenta que se trate de obras prácticas y asequibles desde el punto de vista de su lectura, y que tengan una total disponibilidad para los estudiantes.

Por tanto, hemos optado por títulos que están disponibles en formato electrónico en el catálogo de la Biblioteca de la Universidad de Granada. Esto conlleva dos ventajas: por una parte, el coste cero para el estudiante, y por otra la disponibilidad que nos evita las restricciones típicas de un número limitado de copias en papel.

Dado que están en su último curso de grado, es conveniente que se maneje la bibliografía en inglés, tal y como se ha seleccionado. Todos los libros se basan en ejemplos y proyectos con la placa Arduino, por lo que son prácticos y nos permiten ilustrar ejemplos teóricos e inspirar diseños para el laboratorio de prácticas.

Adicionalmente, se ofrecerá una selección de títulos para profundizar de manera opcional (protocolos, fundamentos, etc.). [2, 3, 4]

Referencias

1. Universidad de Granada. Tecnologías Emergentes, guía docente. Disponible on-line en http://grados.ugr.es/informatica/pages/infoacademica/guias_docentes/optesp/tecnologiasemergentes?pagina_pdf=1 (2013)
2. Barfield, W., Caudell, T, (Eds.). Fundamentals of Wearable Computers and Augmented Reality. CRC Press, New Jersey (2001)
3. Sohraby, K. Miloni, D, Znati, T. Wireless Sensor Networks. Technology, protocols and applications. Wiley, New Jersey (2007)
4. Akay, M. Handbook of Neural Engineering. Wiley, New Jersey (2007)
5. Massachusetts Institute of Technology. MIT Technology Review. Disponible on-line en <http://www.technologyreview.com/> (2013)
6. Arduino. Placa "LillyPad". Disponible on-line en <http://arduino.cc/es/Main/ArduinoBoardLilyPad> (2013)
7. Digi, Inc. Módulos XBee. Disponible on-line en <http://www.digi.com/xbee/> (2013)
8. Neurosky Inc. MindSet. Disponible on-line en <http://www.neurosky.com/Products/MindSet.aspx> (2013)
9. Emotiv. EPOC headset. Disponible on-line en <http://www.emotiv.com/apps/epoc/299/> (2013)
10. Romero, S. y A. Guillén. Evaluación y Autoevaluación Continuas Automatizadas Mediante Herramientas de Software Libre. Actas del I Congreso de Docencia Universitaria. Universidad de Vigo. pp. 259-264 (2009)
11. H. Andrade, Y. Du, Student perspectives on rubric-referenced assessment, in Practical Assesment, Research & Evaluation, vol. 10, n°3, pp. 1-11. (2005)
12. Olsson, T. Arduino wearables. Apress. Berkeley, Calif., (2012)
13. Faludi, R. Building wireless sensor networks. O'Reilly. Beijing, (2011).
14. Igoe, T. Making things talk., O'Reilly. Beijing (2007).
15. Karvinen, T., & Karvinen, K. Make a mind-controlled Arduino robot. Sebastopol, CA, O'Reilly. <http://proquest.safaribooksonline.com/?fpi=9781449312800>. (2012)

ANEXO (en páginas siguientes): Ejemplo de Guía de Trabajo Autónomo de Tecnologías Emergentes.

ANÁLISIS DE TECNOLOGÍAS EMERGENTES

GUÍA DE TRABAJO AUTÓNOMO
DEL TEMA 1

GRADO EN INGENIERÍA
INFORMÁTICA
4º CURSO

Cómo usar esta guía

Como estudiante de cuarto curso de carrera, tu acceso al mundo profesional se aproxima. A lo largo de la carrera no sólo debes haber adquirido una serie de conocimientos técnicos para el ejercicio de tu profesión, sino que además debes haberte transformado en una persona competente. Esto quiere decir que no debes sólo ser un erudito en Ingeniería Informática, sino que además debes poseer una serie de capacidades (“competencias”) que te permitirán desempeñar tu trabajo con éxito (trabajo en equipo, capacidad de comunicación, etc.).

Una de estas competencias esenciales es la capacidad de trabajar de manera autónoma y de seguir formándote por tus propios medios una vez que abandones el ámbito académico. Es lo que se denomina “Life-long Learning”, o aprendizaje permanente a lo largo de la vida. Esta competencia es especialmente relevante en nuestra profesión, dada la velocidad con que aparecen nuevos procesadores, lenguajes, dispositivos...

Esta guía te servirá para adquirir de manera autónoma conocimientos que te permitirán afrontar con éxito el estudio del Tema 1 de la asignatura: Análisis de Tecnologías Emergentes. Si sigues los pasos que se detallan en la guía y realizas las actividades, tu éxito en la adquisición de estos conceptos está asegurado. Si además te quedas con ganas de más, haz también las actividades que se marcan como opcionales.

Oigo y olvido. Veo y aprendo. Hago y entiendo.

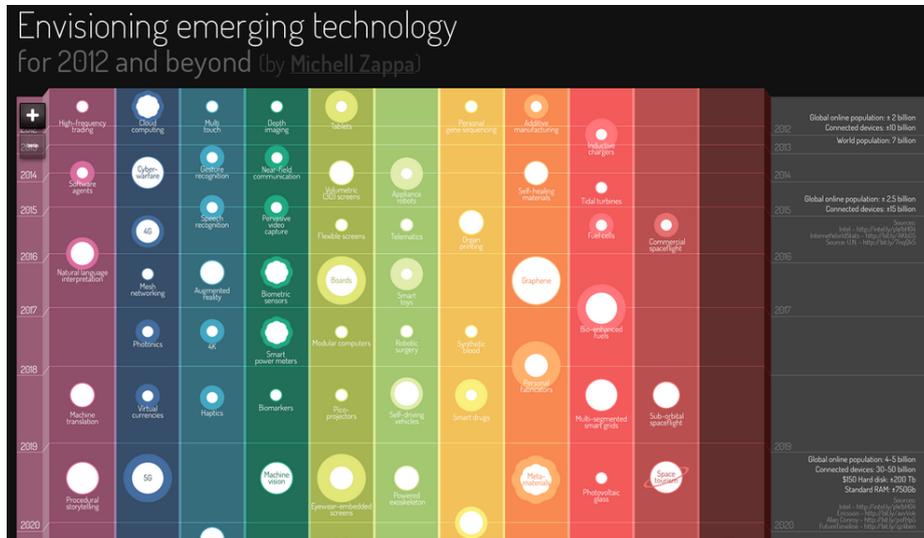
- Confucio, Filósofo chino (551 AC-478 AC).

¿Qué vamos a estudiar en este tema?

A estas alturas de carrera, ya en el último curso, has estudiado multitud de técnicas, métodos, sistemas y herramientas que conforman el cuerpo de esta materia de conocimiento que es la Ingeniería Informática. Si echases un vistazo a planes de estudio anteriores al tuyo, observarás que hay materias que se estudiaban y que tú ya no estudias, y temas nuevos que tus antecesores no tuvieron que estudiar y que a ti te ha tocado aprender. Esta evolución se debe, lógicamente, a que nuestra carrera está en constante renovación (en consonancia con la ciencia y con el mercado), y en el transcurso de pocos años algunas tecnologías llegan a su fin, y otras nuevas surgen.

En este tema vamos a familiarizarnos con las llamadas “Tecnologías Emergentes”. En primer lugar, reflexionaremos sobre el concepto de tecnología en sí, y sobre el ciclo de vida de una tecnología. Analizaremos los factores que influyen en el desarrollo de una tecnología, así como en su éxito o fracaso. Veremos algunos ejemplos de tecnologías informáticas del pasado y apuntaremos a algunas de las nuevas tecnologías que emergen recientemente.

En segundo lugar, y en correspondencia con un primer seminario de la asignatura, aprenderemos a distinguir la calidad de las fuentes de donde obtenemos información acerca de estas tecnologías en fase emergente. Para ello, hablaremos de fiabilidad de las fuentes, analizaremos el tratamiento que se hace de la tecnología en los medios de comunicación, y aprenderemos técnicas para discriminar las informaciones verídicas de las carentes de fundamento.



En este segundo bloque se pretende que desarrolles una competencia de especial valor, no sólo para esta materia o para tu desempeño en la vida profesional, sino para ti mismo en todos los aspectos: el pensamiento crítico.

En el seminario que acompaña al tema 1, aprenderemos a analizar argumentos y sentencias para detectar aquellas que no son válidas (falacias). Esto nos permitirá filtrar las fuentes de donde recoger información sobre tecnologías emergentes. Además, esta competencia te permitirá expresarte mejor, ya que aprenderás también a aplicar estas técnicas a tus propios razonamientos.

El tema continúa distinguiendo fuentes formales (científicas) de informales (medios de comunicación de masas, e Internet), y aprenderemos a realizar búsquedas en ambos ámbitos.

Finalizaremos este tema realizando reflexiones sobre el impacto ético que estas tecnologías potenciales pueden tener.

Los conocimientos que adquieras en este primer tema te serán de gran utilidad en tu trabajo final de la asignatura, en el que realizarás un análisis y presentación ante la clase de una tecnología emergente.

Actividades

1. Busca en el diccionario de la Real Academia el término “tecnología”. ¿en qué se diferencia de “técnica”? ¿y de “ciencia”? Trata de elaborar una definición con tus propias palabras. Pon algunos ejemplos.
2. Mirando atrás en el tiempo, recordarás que han surgido multitud de tecnologías que se prometían superiores a lo vigente, y que sufrieron una “muerte prematura”. Un par de ejemplos podrían ser el sistema de video Beta o el MiniDisc. ¿por qué crees que no triunfaron? Prepara tus argumentos para discutirlos en clase.
3. Localiza otros ejemplos de tecnologías que en su momento emergieron pero fracasaron frente a las tecnologías establecidas.
4. **(opcional)**: a modo de “divertimento”, trata de encontrar videos o anuncios antiguos de dispositivos tecnológicos que hoy nos resultan un tanto ridículos (en YouTube, o Google Imágenes, aunque la “teletienda” es una fuente inagotable).



5. Elabora una tabla con tecnologías informáticas que han terminado su ciclo de vida (“legacy”), indicando el nombre de la tecnología en fase de extinción, nombre de la tecnología nueva que la desplaza, y diferencia de prestaciones (por ejemplo, puerto paralelo vs. Puerto USB).
6. ¿la condición esencial para que una tecnología triunfe en el mercado es que ofrezca mejores prestaciones técnicas que sus competidoras? Si es así, justifícalo. Si no, ofrece algún ejemplo de tecnología que, siendo superior a las demás, no ha triunfado (¿por qué?).
7. Una vez realizadas las actividades anteriores, realiza una lectura comprensiva del bloque 1 de transparencias, disponibles en la plataforma virtual. Trata de relacionar los resultados de las actividades que has realizado con los conceptos que se recogen en el tema. Presta especial atención a los factores que influyen en el desarrollo de una tecnología.
8. Escoge una de las tecnologías que hayas aprendido hace poco en la carrera (puede ser un lenguaje de programación, un nuevo tipo de circuito, o una técnica). Trata de averiguar el origen, o la primera vez que se menciona esta tecnología en un documento (pista: intenta localizar fuentes escritas por los inventores).
9. Lee el bloque 2 de transparencias, dedicado a las fuentes de información formales (revistas científicas, bases de datos, patentes) e informales (radio, televisión, prensa, blogs, páginas web y enciclopedias colaborativas).
10. Pon un ejemplo de cada una de las fuentes mencionadas anteriormente.
11. Para una tecnología cualquiera, localiza una fuente informal y otra formal que la describan (pista: localiza primero la noticia, y luego trata de ver qué hay de cierto consultando la fuente científica). ¿qué diferencias observas?
12. **(opcional)** Echa un vistazo al blog “Malaprensa”. Dispone de un apartado en el que se detectan errores de conceptos, datos científicos o mala interpretación estadística, e incluyen la correspondiente explicación: <http://www.malaprensa.com>
13. Estudia el último bloque de transparencias del tema 1, dedicados al pensamiento crítico, análisis de argumentos y detección de falacias. En el ámbito científico-tecnológico, puede ser útil usar estos “diez mandamientos” para distinguir la ciencia de los camelos (pseudociencias):
 - I. Basaras tus conocimientos en la evidencia.

La evidencia en su forma más básica es una observación, así que esta regla implica que las conclusiones deben ser basadas en observaciones. como un detective en la escena del crimen, o en un juicio, las conclusiones están soportadas en evidencias

- II. Medirás objetivamente y no supondrás subjetivamente
- III. Tus afirmaciones serán respaldadas por evidencia (En otras palabras, decir que algo es un hecho no lo hace un hecho)
- IV. Usarás muestras de gran tamaño
- V. Tus experimentos serán a ciegas
- VI. Tus Experimentos tendrán grupo control
- VII. Citarás tus fuentes de información
- VIII. Tus fuentes de información serán confiables, verificables y amparadas por la evidencia
- IX. La opinión no es un hecho
- X. No darás falso testimonio ni mentiras (No harás trampa)

(del blog <http://razonesyfalacias.blogspot.com.es/p/ciencia-metodo-y-mbe.html>).

- 14. Para el caso de la noticia de la actividad 11, localiza falacias que no se correspondan con la fuente original. Explica por qué no son afirmaciones válidas.
- 15. Selecciona una tecnología emergente (por ejemplo, la identificación por radiofrecuencia), y reflexiona sobre los riesgos éticos que presenta. Prepara argumentos para discutirlos en clase.
- 16. **(opcional)** Un par de lecturas recomendables para analizar información científica:
 - Cómo mentir con estadísticas (Darrel Huff, Ed. Crítica, 2011)
 - El mundo y sus demonios (Carl Sagan, Ed. Planeta, 2005)

Vivimos en una sociedad profundamente dependiente de la ciencia y la tecnología y en la que nadie sabe nada de estos temas. Ello constituye una fórmula segura para el desastre.

- Carl Sagan, astrónomo (1934-1996).

Implementación de algoritmos en hardware: de la puerta NAND al bucle FOR

Javier Díaz Alonso

Departamento de Arquitectura y Tecnología de Computadores
E.T.S.I.I.T., Universidad de Granada, España

jda@ugr.es

Resumen. En este artículo se describe el trabajo a realizar en la asignatura de Implementación de algoritmos Hardware dentro de las titulaciones de grado en Ingeniería Informática (especialidad de Ingeniería de Computadores) y de grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada. Motivaremos y contextualizaremos la asignatura, explicando a los estudiantes por qué este conocimiento es útil y conveniente como parte de su especialización. Describiremos los contenidos a impartir, mostrando que el diseño de sistemas hardware puede hacerse a diferentes niveles de abstracción y, particularmente, usando lenguajes de programación convencionales. Finalmente mostraremos la metodología didáctica y métodos de evaluación propuestos, completando con ello la descripción global de esta asignatura.

Palabras Clave: Docencia en Ingeniería de Computadores, diseño de Hardware, SoC, Aprendizaje basado en proyectos.

Abstract. This article describes the work to be done in the course Hardware Implementation of algorithms within undergraduate degrees in Computer Science (Computer Engineering specialization) and Telecommunications Technology Engineering at the University of Granada. We motivate and contextualize the subject, explaining to students why this knowledge is useful and appropriate as part of their specialization. We describe the contents to impart, showing that hardware system design can be made from different attraction levels and particularly, using conventional programming languages. Finally we show teaching the methodology and evaluation methods proposed, thereby completing the comprehensive description of this course

Keywords: Computer engineering teaching, hardware design, SoC, project-based learning

1 Introducción

“El hardware es muy complicado. Requiere conocer muchos detalles, la mayoría de muy bajo nivel (voltajes e interfaces eléctricos, temporización de señales, principios físicos, etc.). Además nosotros no podemos fabricar chips (saldría muy caro) e incluso comprarlos para hacer un placa de circuito impreso requiere inversión o acceso a maquinaria difícil de obtener. Por todo ello la mejor opción es comprar hardware comercial. Este debe ser flexible y escalable en prestaciones de forma tal que nuestro objetivo sea su personalización mediante la programación del software asociado.”

Las frases anteriores bien podrían describir la opinión de muchos estudiantes de las titulaciones de grado en Ingeniería Informática (especialidad de Ingeniería de Computadores) y de grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada. El hardware se antoja como algo “feo y complejo”. Operaciones que hacemos todos los días como leer un valor del teclado o una memoria USB requieren horas sino semanas de trabajo para conseguir estos objetivos si pensamos en diseñar el hardware asociado. Si a esto le añadimos la confusión de que el termino hardware puede representar muchos conceptos diferentes (chips, tarjetas PCBs, elementos mecánicos, etc..) y que muchos autores de textos docentes no ayudan demasiado (ver por ejemplo [1] donde se dice que “los sistemas operativos ocultan el hardware feo con abstracciones hermosas”), se crea el clima perfecto para desmotivar al estudiante a profundizar en estos contenidos, situación que no es fácil de superar.

Pero, ¿qué ocurriría si el hardware se pudiera describir mediante programación en C/C++? ¿Y si este pudiera cambiarse, programarse, de la misma forma que se hace con el software convencional? ¿Y si la división, la frontera entre hardware y software hubiera, hoy en día, desaparecido? ¿Y si en vez de semanas pudiéramos tener sistemas hardware complejos en menos de 1 hora? Además, ¿qué hacemos cuando los miles de millones de transistores que traen los dispositivos de cómputo actuales no son suficientes para cumplir con los requisitos de nuestro proyecto?

En la asignatura de Implementación de Algoritmos en Hardware (IAH) tratamos de dar respuesta a estas preguntas y otras relacionadas. Centrándonos principalmente en el campo de los sistemas empuotrados, trabajamos en el análisis y diseño hardware entendiendo como tal los elementos de computación y control existentes en un chip, típicamente programable como por ejemplo una FPGA. Mostramos que simplemente usando tarjetas para desarrollo de prototipos, podemos diseñar sistemas empuotrados complejos sin más que reprogramar la funcionalidad de un único chip, diseñando/modificando los elementos software o hardware del sistema, eliminando la férrea distinción que existía entre ambos aspecto en el pasado. Como resultado, veremos que una buena división de los elementos hardware y software del sistema permite conseguir aumentos significativos de prestaciones o por ejemplo reducciones de consumo considerables, aspectos claves para el desarrollo de los sistemas empuotrados actuales.

La asignatura de IAH parte de conocimientos generales del estudiante sobre sistemas digitales y la estructura/arquitectura de computadores. A partir de ellos veremos que es posible no sólo desarrollar nuevos sistemas empotrados de altas prestaciones para problemas de uso específico, sino que también, gracias a este conocimiento podemos programar los procesadores sacando máximo provecho de los potentes elementos hardware incluidos actualmente en ellos.

A lo largo de la asignatura utilizamos el concepto de System-on-Chip (SoC) y su presencia en la vida cotidiana en múltiples dispositivos como teléfonos, tabletas, routers de comunicaciones o cámaras de videovigilancia. Del análisis de estos dispositivos podemos identificar la gran variedad de elementos hardware que contienen los procesadores empotrados actuales, ya sean módulos de comunicaciones, coprocesadores multimedia, coprocesadores aceleradores o controladores de entrada/salida. Cada uno de ellos da respuesta a la solución de un problema específico y el conocimiento de éstos es vital para entender el desarrollo de los sistemas actuales.

Como conclusión, ¿qué aporta a los estudiantes cursar esta asignatura? ¿Para qué el esfuerzo? La realización de esta asignatura capacita a los alumnos a desarrollar eficientemente sistemas empotrados donde la división entre software y hardware sea realizada de forma óptima. Permite seleccionar la plataforma empotrada comercial más adecuada a nuestras especificaciones, de forma que minimicemos los costosos ciclos de rediseño debido a una mala selección de elementos del sistema. Finalmente, el conocimiento adquirido nos permite sacar el mejor provecho de los procesadores empotrados presentes en dispositivos multimedia (tabletas, teléfonos), en sensores inalámbricos y de red, en dispositivos de automoción, aviónica, aeroespacial, etc..., aumentando significativamente gracias a todos estos conocimientos nuestra especialización y valor como ingeniero en el campo de los sistemas empotrados.

2 Implementación de algoritmos en hardware en el contexto de la titulación

La asignatura de IAH se imparte como optativa durante el 4º curso del grado en Ingeniería Informática (especialidad de Ingeniería de Computadores). Además, se ofrece como optativa de otras titulaciones del grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada.

Tal y como se establece en la memoria Verifica del título de Ingeniería Informática de la Universidad de Granada [2], junto con la asignatura del Tecnologías emergentes pertenece a los Complementos de Sistemas de Cómputo para Aplicaciones Específicas y tiene una carga lectiva de 6 créditos ECTS.

Como se desprende de los objetivos formativos descritos en la tabla 1 y de las competencias descritas en la tabla 2, la asignatura pretende capacitar al estudiante en las tareas de análisis y diseño de elementos hardware de un sistema de computación, así como ser capaz de evaluar y seleccionar los elementos existentes. Es por ello, que el enfoque de la asignatura sigue un planteamiento global, evaluando los posibles soluciones de un problema dado de forma integral, seleccionando un particionamiento software/hardware que permita conseguir la solución de diseño óptima para un

problema teniendo en cuenta restricciones de prestaciones, potencia, coste, fiabilidad o tiempos de desarrollo.

Destaquemos que en los grupos de estudiantes de esta asignatura es posible la participación de alumnos de distintos grados, informática y telecomunicaciones. Lejos de ser un problema y como se expondrá en la sección 4, el enfoque metodológico variado, con grupos de trabajo y aprendizaje basado en proyectos puede/debe beneficiarse significativamente de esta diversidad de experiencias y permitir encontrar/resolver problemas de naturaleza más interdisciplinar.

En lo referente a su relación con otras asignaturas, un diagrama de bloques que muestra su relación con otras asignaturas de la memoria del grado en Ingeniería Informática es mostrado en la Figura 1.

Tabla 1. Objetivos formativos de IAH.

• Describir y aplicar metodologías de diseño de circuitos digitales para sistemas de uso específico.
• Aplicar técnicas de análisis de prestaciones (precisión, recursos y potencia de cómputo) para la identificación de elementos que limitan el sistema.
• Identificar las herramientas y metodologías de diseño más adecuadas según la aplicación y sus especificaciones.
• Conocer técnicas de diseño de módulos hardware de altas prestaciones (coprocesadores o periféricos) para sistemas integrados.
• Aplicar técnicas de máquinas de estado algorítmicas.
• Conocer técnicas de diseño de circuitos de operaciones matemáticas complejas. Elegir el tipo de representación de los datos acorde a las especificaciones de precisión, recursos y prestaciones.
• Identificar y reconocer las necesidades funcionales que justifican el desarrollo de hardware de propósito específico en distintos campos de aplicación.
• Aplicar metodologías de adaptación de algoritmos para su implementación en hardware de propósito específico.
• Aplicar técnicas avanzadas de compartición de recursos y optimización de prestaciones (segmentación de cauce, diseño superescalar, etc.).
• Evaluar costes y prestaciones de módulos hardware para aplicaciones específicas.
• Identificar dispositivos y plataformas integradas para aplicaciones específicas, sus restricciones y campos de aplicación.

Tabla 2. Competencias de la asignatura IAH

- **IC1.** Capacidad de **diseñar y construir sistemas digitales**, incluyendo computadores, sistemas basados en microprocesador y sistemas de comunicaciones.
- **IC2.** Capacidad de desarrollar **procesadores específicos y sistemas empotrados**, así como desarrollar y **optimizar el software** de dichos sistemas.
- **IC5.** Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuadas para el soporte de aplicaciones empotradas y de tiempo real.
- **E4.** Capacidad para **definir, evaluar y seleccionar plataformas hardware** y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- **E9.** Capacidad para **resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad**. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

La asignatura requiere conocimientos básicos sobre sistemas digitales y estructura de los sistemas de computación. Estos conocimientos son parte los contenidos tratados en la asignaturas como Tecnología y Organización de Computadores, Estructura de Computadores y Arquitectura de Computadores

Ya en el contexto de la especialidad en Ingeniería de Computadores, la asignatura tiene una fuerte relación con las obligatorias de la especialidad de Sistemas empotrados y Desarrollo de hardware digital. La primera supone un complemento natural a los contenidos aquí expuestos al tratar, entre otros, los contenidos necesarios para desarrollar un los elementos software de programación de los procesadores desde la programación de bajo nivel hasta el nivel de sistema operativo. En lo referente a la asignatura de Desarrollo de hardware digital, ésta proporciona un puente entre los conocimientos de tecnología de computadores básicos y los conocimientos abordados en esta asignatura. De esta forma la conexión entre los distintos niveles de descripción de un sistema digital está completa.

Cabe por último destacar otras asignatura de la especialidad de Ingeniería de Computadores que, si bien no tienen una conexión tan directa, se benefician/complementan los contenidos aquí expuestos. Por ejemplo en la asignatura de Sistemas con microprocesadores se centra en la descripción y programación de un subconjunto de los elementos que veremos en IAH. Tecnologías emergentes describe numerosos sistemas de aplicaciones específicas donde la se hace vital el desarrollo de elementos hardware a medida tal y como los descritos en esta asignatura. Finalmente, en la asignatura de diseño de sistemas electrónicos se describen a bajo nivel los componentes electrónicos utilizados para el diseño de los bloques descritos en IAH.

De forma similar a lo descrito para el grado de Ingeniería Informática, en el grado en Ingeniería de Tecnologías de Telecomunicación, de forma destacada en el marco

de la especialidad de sistemas electrónicos, la asignatura IAH aporta una visión de alto nivel y aplicada de los bloques básicos analizados en otras asignaturas de la especialidad. De esta forma la asignatura proporciona en este grado una visión aplicada y funcional de los elementos hardware que se utilizan en el diseño de un sistema empotrado. Como se explica en la sección 3, la utilización de lenguajes de descripción hardware (HDLs) y de dispositivos reconfigurables (FPGAs) proporcionan una notable flexibilidad para el diseño rápido de sistemas electrónicos y, en concreto, para la implementación de algoritmos en hardware de propósito específico.

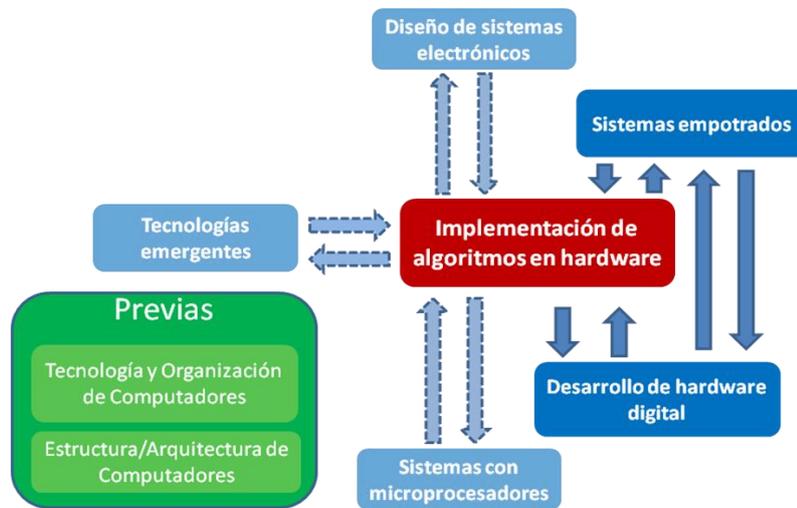


Figura 1. Relación de IAH con otras asignaturas del grado de Ingeniería Informática.

3 Contenidos de la asignatura Implementación de algoritmos en hardware

Los objetivos y competencias descritos en la tablas 1 y 2 toman forma con una selección de contenidos que es la descrita en la tabla 3. En lo referente a los contenidos de teoría, la asignatura comienza con un tema de contenidos generales para luego tener los temas 2 y 3 dedicados a los contenidos más propiamente relacionados con el cuerpo técnico de la materia. El 4 es un tema de carácter más aplicado y divulgador y que pretender conectar los contenidos más teóricos descritos en los temas anteriores con contenidos de carácter práctico de la asignatura, analizando para ello sistemas comerciales o de investigación existentes y/o en desarrollo actualmente.

Tabla 3. Contenidos de la asignatura Implementación de algoritmos en hardware

TEORÍA:

- **Tema 1: Introducción y conceptos generales**
 - 1.1 Motivación y objetivos de la implementación de algoritmos en hardware.
 - 1.2 Tecnologías y dispositivos.
 - 1.3 Conceptos generales (niveles de abstracción, síntesis automática, módulos IP, SoC, NoC, codiseño hardware-software).
 - 1.4 Lenguajes de descripción y herramientas de diseño.
- **Tema 2: Metodologías de diseño e implementación de algoritmos en hardware**
 - 2.1 Sistemas autónomos, periféricos y coprocesadores.
 - 2.2 Diseño a nivel de sistema. SoC y particionamiento hardware software. Metodología de diseño basada en plataformas..
 - 2.3 Diseño RTL. Diseño basado en máquinas algorítmicas.
 - 2.4 Tipos de aritméticas para implementación de algoritmos.
 - 2.5 Compromisos de diseño: consumo de recursos, prestaciones, precisión, potencia, coste, tiempo de diseño.
 - 2.6 Técnicas de diseño y optimización: compartición de recursos, planificación temporal, segmentación de cauce, diseño superescalar, diseño para bajo consumo.
- **Tema 3: Hardware para operaciones específicas**
 - 3.1. Técnicas de diseño de funciones aritméticas y trigonométricas.
 - 3.2. Controladores de memoria, buses, E/S.
 - 3.3. Arquitecturas de procesamiento SIMD/MIMU. Arquitecturas basadas en multicores/NoCs.
 - 3.4. Otras arquitecturas de interés (registros LFSR, procesadores sistólicos, etc..).
- **Tema 4: Ejemplos de sistemas y aplicaciones.**
 - 4.1 Identificación de requisitos de hardware de propósito específico en aplicaciones de comunicaciones, procesamiento de imágenes y video, audio, control y criptografía. 5.1 Aplicaciones para procesamiento de audio.
 - 4.2. Ejemplos para procesamiento de imágenes y video.
 - 4.3 Ejemplos para comunicaciones.
 - 4.4 Ejemplos para control.
 - 4.5 Ejemplos para criptografía.
 - 4.6. Ejemplos para supercomputación y aceleración de procesos.

SEMINARIOS:

- **Seminario práctico 1:** Introducción al entorno de desarrollo y lenguaje VHDL.
- **Seminario práctico 2:** Implementación de algoritmos en aritmética en punto fijo: recursos, precisión y prestaciones.
- **Seminario práctico 3-4:** Introducción a los lenguajes de descripción de hardware basados en C/C++.
- **Seminarios prácticos 5:** Ejemplos de arquitecturas para comunicaciones, procesamiento de audio/video y criptografía.

PRÁCTICAS

- **Practica 1:** Diseño e instanciación de módulos IP para aceleración de operaciones matemáticas/procesamiento digital de señales.
- **Practica 2:** Compresión JPEG. Análisis de arquitecturas hardware. Estudio de las diferentes arquitecturas, estrategias de diseño y su impacto en el diseño resultante.
- **Practica 3.** Implementación en hardware de un algoritmo para aplicaciones específicas.

Respecto a la bibliografía para estos contenidos de teoría, dado que la asignatura requiere una actualización continua de los mismos para reflejar las últimas tendencias existentes en la industria y la academia, buena parte de ellos serán obtenidos de páginas webs de compañías y revistas de carácter industrial. No obstante, para una revisión de los contenidos más clásicos o de índole más teórica, recomendamos la consulta de los trabajos siguientes [4, 5, 6, 8].

La bibliografía de contenidos de índole más aplicada, seminarios y prácticas, será obtenida principalmente de sitios web de proveedores de herramientas, chips, tarjetas, etc.. Destacamos para ello las referencias [9, 10, 11, 12, 13, 14] de donde pueden obtenerse buena parte del material que utilizamos así como información actualizada de las últimas tendencias en la temática.

Como resultado de este enfoque bibliográfico actualizado y de la metodología docente descrita en la sección 4, aproximadamente un 50-60% de los contenidos impartidos son fijos, siendo un 10-20% dependiente de las tendencias actuales de la tecnología y aproximadamente un 30% se determinan cada año en base a los intereses del alumnado matriculado en ese curso.

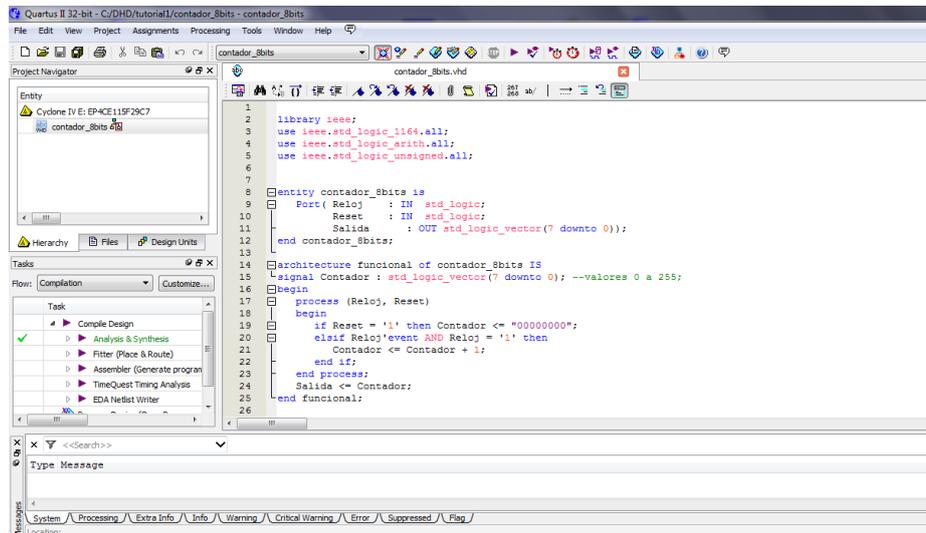


Figura 2. Entorno de trabajo Quartus II de Altera.

Finalmente, en lo referente a los elementos a utilizar en clase, destacar el uso del entorno de programación Quartus II de Altera y la tarjeta DE2-115 [9]. Las imágenes de estos elementos se muestran en las figuras 2 y 3. Como se aprecia en la Figura 3, la tarjeta FPGA utilizada consta de numerosas interfaces y periféricos lo que nos permite desarrollar una gran cantidad de proyectos usando la misma tarjeta.



Figura 3. Tarjeta de desarrollo de prototipos, DE2-115 de Altera.

4 Secuenciación temporal, metodología y evaluación

En los apartados anteriores hemos mostrado los objetivos, competencias y contenidos de la asignatura IAH. De forma simplificada y como resumen podemos decir que la asignatura persigue tres objetivos fundamentales que son:

a) Repasar, dar una visión homogénea a los estudiantes de conceptos básicos relacionados con IAH. Esto se consigue mediante los el tema 1 de teoría y el seminario y práctica 1.

b) Desarrollar los objetivos y competencias de la asignatura. Esto es tratado en los temas 2 y 3, en los seminarios 2-4 y en la práctica 2.

c) Conectar el conocimiento adquirido con los sistemas disponibles actualmente para capacitar al ingeniero en formación en el desarrollo autónomo de los sistemas estudiados y/o en la selección de los mismos. Estos apartados son cubiertos en el tema 4 de teoría, en el seminario 5 y en la práctica 3.

El primer objetivo tiene una duración temporal de 2-3 semanas y se desarrolla con una metodología clásica, con lecciones magistrales pero participativas y con ejercicios muy concretos y específicos.

El segundo objetivo utiliza una metodología mixta basada en lecciones magistrales, aprendizaje basado en proyectos, metodología basad en discusión y metodología

basada en aprendizaje autónomo usando las guías de trabajo autónomo. Tiene una duración temporal de aproximadamente 7-8 semanas.

Finalmente, el último bloque tiene una duración de 4-5 semanas y dota de gran autonomía y poder de decisión a los alumnos. Se centra en las metodologías de aprendizaje basado en proyectos y en el trabajo autónomo del alumno. Esto se materializa mediante la creación de grupos de trabajo que resuelvan en la práctica 3 un problema concreto propuesto por los alumnos y aceptado por el profesor. Además, en los contenidos de teoría se fomentará la participación del alumnado en forma de exposiciones orales y discusiones/debates de artículos de forma que las competencias como IC5 o E9 queden cubiertas. De esta forma, la última parte de la asignatura vendrá determinada por los intereses de los alumnos matriculados y producirá que, como se ha descrito en la sección 3, aproximadamente un 30% de los contenidos sean variables acorde a los intereses de los alumnos.

Esta diversidad de metodologías persigue tanto desarrollar los conocimientos del alumno como sus competencias de carácter práctico. Se complementan con el uso de las tutorías, individuales o grupales como un recurso didáctico más que es planificado para permitir conseguir un correcto resultado de aprendizaje y una buena comunicación profesor-alumno.

Acorde a la diversidad de métodos docentes utilizados, la evaluación docente es también rica en los métodos a utilizar. Se fomenta la evaluación continua de los alumnos. Exámenes, cuestiones en clase, trabajos de exposición oral, realización de prácticas de laboratorio, trabajos libres de carácter práctico, etc.. Muchos métodos son posibles y parte de ellos se concretan dependiendo de los intereses de los alumnos matriculados y previo pacto profesor-alumno. De igual manera, en peso en la evaluación de los diferentes objetos calificables también es pactado entre los alumnos y el profesor. Recordemos que el objetivo de estas evaluaciones es ayudar al proceso de aprendizaje y no el conseguir poner un nota. Por ello, se trata que los estudiantes reciban pronto las valoraciones y que el número de pruebas sea suficiente para informarles del progreso de su aprendizaje.

Con carácter orientativo, un posible esquema de la distribución de la nota en la evaluación podría ser el siguiente:

- Trabajo autónomo y en aula (individual y en grupo) → 30%
- Prácticas guiadas y libres → 40 %
- Pruebas/exámenes → 30%

Se pretende que cada año los alumnos sean parte activa de la evaluación mediante utilización de técnicas de autoevaluación o de evaluación entre iguales. Su uso más o menos intensivo depende del interés de los alumnos y del número de matriculados (si el número supera los 20-25 se hace imprescindible para, sin eliminar pruebas de evaluación, mantener una carga de trabajo del profesor razonable).

Por último destacar que aunque se trata de utilizar una evaluación continua del alumnado, ya sea porque estos no han podido asistir o bien porque no han superado las pruebas realizadas, existe una última prueba consistente en un examen teórico-práctico global para permitir superar la asignatura a aquellos que así lo requieran.

5 Conclusiones

Este trabajo ha tratado de describir el trabajo que se desarrolla en la asignatura de Implementación de algoritmos en hardware. Hemos visto una motivación general en la sección 1, los objetivos y competencias en la sección 2, los contenidos en la sección 3 y la secuenciación de los mismos, metodología didáctica y evaluación en la sección 4. Con ello se ha descrito de forma global la asignatura y su forma de impartición.

Es importante destacar que esta asignatura tiene un carácter eminentemente práctico. La utilización de lenguajes de descripción de hardware como VHDL y sobre todo los basados en lenguajes de alto nivel como C/C++ son clave para acercar a los alumnos a esta materia y base de la productividad de los diseños a realizar.

La realización de esta asignatura ayuda a mejorar la capacitación de los estudiantes en un campo en continua expansión como es el de los sistemas empotrados. No sólo es gracias a la mejora de sus capacidades de diseño y análisis de estos sistemas, sino también porque este conocimiento permite una selección adecuada de las plataformas empotradas a utilizar y su programación optimizada, capacidades claves para su futuro ejercicio profesional.

Finalmente concluir incidiendo en a gran libertad de decisión que tienen los alumnos en la asignatura, permitiendo metodologías de trabajo en grupo muy similares a las que utilizarán en la industria durante el ejercicio de su carrera profesional y con posibilidad de seleccionar parte de los contenidos de la asignatura en base a sus propios intereses e inquietudes.

Referencias

1. Tanenbaum, Andrew S.: Sistemas operativos modernos. 3ed. Prentice Hall, pag. 5 (2003)
2. Memoria Verifica del título de Ingeniería Informática de la Universidad de Granada. Disponible online en el siguiente enlace:
<http://grados.ugr.es/informatica/pages/infoacademica/archivos/verificaingenieriainformatica/%21>
3. Steve Kilts, “Advanced FPGA Design: Architecture, Implementation, and Optimization”. John Wiley and Sons, 2007.
4. Clive Maxfield, “The design warrior’s guide to FPGAs”, Elsevier 2004.
5. A. Rushton: "VHDL for Logic Synthesis", John Wiley and Sons, 2001.
6. D.L. Perry, "VHDL programming by example", McGraw-Hill, 2002. (Recurso electrónico).
7. J.O. Hamblen, T. S. Hall, M. O. Furman: “Rapid Prototyping of Digital Systems : SOPC Edition”, Springer 2008.
8. J. Bhasker, “A SystemC Primer”, Star Galaxy Publishing (2002).
9. Altera, fabricantes de dispositivos reconfigurables. Online en: <http://www.altera.com>
10. Xilinx, fabricantes de dispositivos reconfigurables. Online en: <http://www.xilinx.com>
11. Principal portal de descargas de código HDL Opencores. Online:
<http://www.opencores.com>
12. Open hardware repository. Online: <http://www.ohwr.org>
13. Promotores iniciativa SystemC. Online: <http://www.accellera.org/home/>
14. Sitio web sobre empresas, foros, eventos, etc.. Online: <http://www.fpgacentral.com/>

Complementos para Informática Industrial del perfil de Ingeniería de Computadores del Grado de Informática de la UGR: Controladores Lógicos Programables

M. Damas, O. Baños, G. Olivares, F. Gómez

Departamento de Arquitectura y Tecnología de Computadores. ETSI Informática y de Telecomunicación. Universidad de Granada.
{mdamas, oresti, gonzalo, frgomez}@ugr.es

Resumen. En este trabajo se presentan tanto la motivación como la metodología docente de la asignatura Controladores Lógicos Programables, una optativa de la materia de complementos para Informática Industrial del perfil de Ingeniería de Computadores del Grado de Informática de la ETSIIT de Granada, presentada en el marco de las III Jornadas de Coordinación Docente y de Empresas organizadas por el Departamento de Arquitectura y Tecnología de Computadores. Concretamente, en este trabajo se muestra el porqué de esta materia y asignatura, cómo se ha organizado para lograr los objetivos propuestos en la guía docente, la metodología y herramientas que se van a utilizar en la parte práctica de la asignatura, y finalmente las principales conclusiones de este trabajo.

Palabras Clave: Informática Industrial, Controladores Lógicos Programables, motivación del alumnado, laboratorio virtual, Ingeniería de Computadores.

Abstract. This paper shows the motivation and teacher methodology of the Programmable Logic Controller course, an elective of the Industrial Computer Complement subject of the Computer Engineering profile in Computer Sciences Grade of Granada ETSIIT, presented at the III Conference on Educational Coordination and Enterprises, organized by the Department of Computer Architecture and Computer Technology. Particularly here we show why this subject and course, how it is organized to achieve the objectives in the teaching guide, the methodology and tools to be used in the practical part of the course, and finally the main conclusions of this work.

Keywords: Industrial Computing, Programmable Logic Controllers, student motivation, virtual laboratory, Computer Engineering.

1 Introducción

La puesta en marcha de nuevos Planes de Estudios en los que el Departamento de Arquitectura y Tecnología de Computadores (ATC) de la Universidad de Granada imparte asignaturas (Grados en Telecomunicaciones, Informática, Físicas, Electrónica industrial y Químicas), y la existencia en alguno de estos planes (Grado en Ingeniería Informática) de perfiles como el de Ingeniería de Computadores (IC), con una participación mayoritaria del mencionado Departamento, ponen de manifiesto la necesidad de desarrollar tanto actividades de coordinación de la docencia de las nuevas asignaturas, como de difusión de las perspectivas laborales asociadas a las competencias de esos perfiles.

Una de estas actividades es la organización anual de las Jornadas de Coordinación Docente y Empresarial del Departamento de Arquitectura y Tecnología de Computadores (JCDE) [1], y precisamente este trabajo se enmarca dentro de la III edición de estas jornadas, enfocadas en esta ocasión a coordinar y difundir las asignaturas optativas que imparte el Departamento de Arquitectura y Tecnología de Computadores en el Grado de Informática de la ETSIT de la Universidad de Granada.

Concretamente, la optativa que se presenta aquí está incluida en la materia de Complementos de Informática Industrial del Perfil de Ingeniería de Computadores del Grado de Informática, junto con otra asignatura denominada Informática Industrial (ver figura 1). Ambas asignaturas tienen 6 créditos cada una y por tanto en total son 12 créditos los dedicados a complementos de Informática Industrial. Con este trabajo se pretende, en primer lugar, justificar la inclusión de esta materia como optativa en el Grado de Ingeniería Informática, ya que a priori parecería que las competencias que se puedan adquirir con ella pertenecen más al ámbito de la Ingeniería Industrial.



Figura 1. Complementos de Informática Industrial en el perfil de Ingeniería de Computadores en el Grado de Informática

A continuación, y una vez argumentados los motivos por los que consideramos apropiado impartir dicha materia y en especial la asignatura Controladores Lógicos Programables en el Grado de Ingeniería Informática, también se muestra la metodología docente que se va a seguir y los equipos y herramientas que se van a utilizar en la parte práctica de dicha asignatura. Finalmente se presentan las conclusiones más importantes de este trabajo.

2 ¿Por qué esta materia y asignatura?

2.1 ¿Por qué esta materia?

En primer lugar, hay que decir que la informática y concretamente la ingeniería de computadores cada vez está más presente en el ámbito de la automatización industrial. Por ejemplo, las arquitecturas de los controladores que se utilizan en la industria cada vez tienen mayores prestaciones, he incluso nos encontramos con configuraciones paralelas y redundantes, por no hablar de que los paradigmas actuales, como por ejemplo el Cloud Computing, aunque más lentamente también están llegando al entorno industrial.

Además, las herramientas de programación y configuración de los equipos de automatización (PLC, DCS, PAC, etc.) [2], así como algunas aplicaciones específicas como los SCADA [3], el estándar de comunicaciones OPC [4] o las herramientas MES [5] que se utilizan actualmente en la industria, son cada vez más complejas y con mayores funcionalidades.

Por lo tanto, es de suponer que un ingeniero informático estará más familiarizado a herramientas y aplicaciones similares a estas, y seguro que les sacará un mayor rendimiento. En la figura 2 se muestran ejemplos de estas configuraciones y herramientas en el ámbito industrial que cada vez son más sofisticadas.



Figura 2. Ejemplos de equipos y herramientas actuales en la automatización.

2.2 ¿Por qué esta asignatura?

Una vez introducida brevemente la motivación para cursar esta materia en el Grado de Informática, a continuación nos vamos a centrar en dar algunos motivos más por los que la asignatura Controladores Lógicos Programables se debe impartir dentro de esta materia:

- En primer lugar señalar que el concepto de PLC está muy extendido y su modelo en cuanto a diseño, programación y configuración es seguido por otros equipos similares: DCS, PAC, RTU, Slot-PLC, Soft-PLC, etc.
- Los PLC son los sistemas más utilizados en la automatización de proceso industriales, como lo demuestra la gráfica siguiente (figura 3) donde se puede observar el % de estos sistemas instalados.

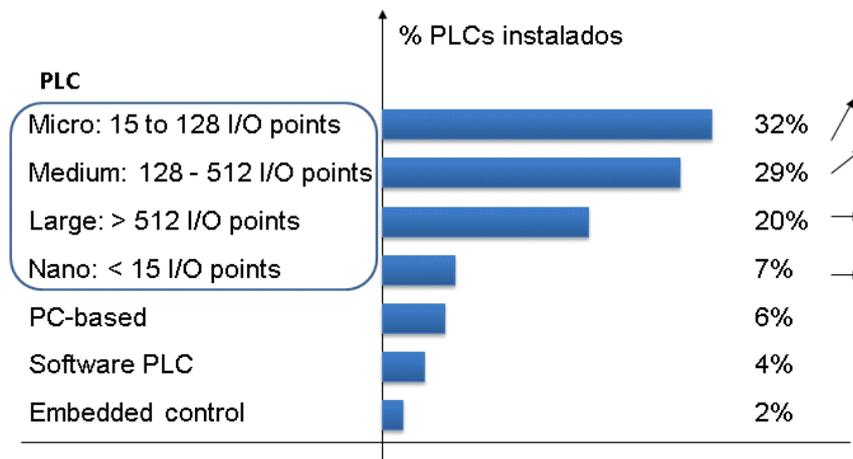


Figura 3. Utilización de sistema de control en la industria
(Fuente: Control Engineering, Reed Research, 2009).

- Los PLC además se utilizan en todo tipo de industrias: automatización de fábricas, procesos y edificios, industrias de generación y transporte de energía, etc.
- Los Ingenieros Informático con estos conocimientos complementan muy bien a los Ingenieros Industriales, como se ha podido constatar en empresas cercanas y pertenecientes al sector de la automatización industrial como ICR S.A., Akron, Ingenion Soluciones, etc.
- Otro motivo por el que impartir estas asignaturas es el de potenciar el área de Ingeniería de Sistemas y Automática del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada.
- Y finalmente indicar también que una asignatura optativa similar en la ingeniería informática a extinguir ha tenido hasta ahora una buena aceptación por parte de los alumnos de esta titulación, tal como se puede ver en la siguiente gráfica (figura 4), donde se puede apreciar que en término medio se han matriculado en torno a 35 alumnos.

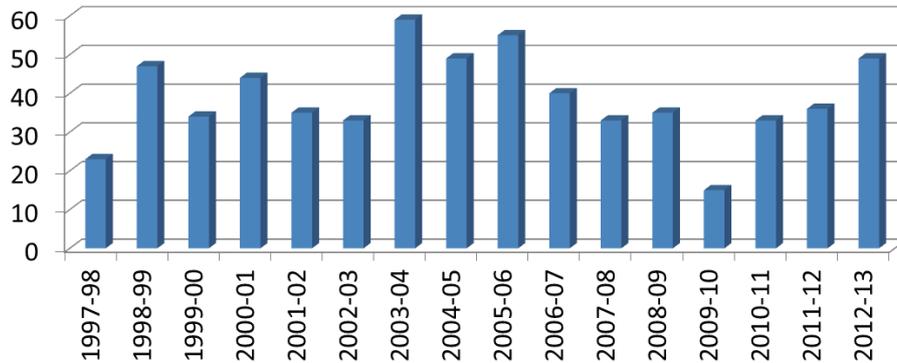


Figura 4. Número de alumnos matriculados en la asignatura PLC de la Ingeniería Informática a extinguir

3 Temario Teórico y Práctico: Guía Docente

En la guía docente elaborada para la asignatura Controladores Lógicos Programables aparecen los siguientes objetivos expresados como resultados de aprendizaje:

- Definir los conceptos básicos sobre automatismos en la industria.
- Conocer las arquitecturas típicas que se utilizan en el control de procesos industriales.
- Distinguir las características que diferencian a un PLC del resto de arquitecturas de control.
- Identificar las distintas partes que conforman la arquitectura interna de un PLC.
- Comprender el ciclo de funcionamiento interno de un PLC y su vinculación con el control en tiempo real.
- Diseñar un PLC teniendo en cuenta sus posibles configuraciones e interfaces de Entrada/Salidas y Específicas.
- Programar un PLC con los diferentes lenguajes que propone el estándar IEC 61131-3.
- Utilizar diferentes entornos de programación de PLC.
- Identificar el papel de los PLC en las redes de comunicaciones industriales.
- Conocer las redes comerciales más usadas en la automatización de procesos industriales.

En base a estos objetivos se ha planteado el siguiente temario teórico y práctico para esta asignatura, teniendo en cuenta que los temas teóricos y los seminarios necesarios para poder impartir las prácticas estén correctamente temporizados, es decir, que se impartan antes o como mucho al mismo tiempo.

TEMARIO TEÓRICO:

Tema 1. Introducción al Control Industrial

- 1.1. Sistemas de Control
- 1.2. Automatismos cableados y programables
- 1.3. Arquitecturas de control: Unidades terminales remotas (RTU), Controladores lógicos programables (PLC), Controladores industriales, Ordenadores industriales (IPC), Controladores de Automatización Programables (PAC), Control basado en PC (Slot-PLC, Soft-PLC), etc.
- 1.4. Centros de control y software SCADA
- 1.5. Fabricación integrada por computador (CIM)

Tema 2. Diseño de Automatismos

- 2.1. Automatismos combinacionales y secuenciales
- 2.2. Grafo de control etapa-transición (GRAFCET)
- 2.3. Puestas en marcha y paradas
- 2.4. Diseño de automatismos de procesos continuos

Tema 3. Controlador Lógico Programable

- 3.1. Arquitectura interna del PLC
- 3.2. Ciclo de funcionamiento y control en tiempo real
- 3.3. Configuración del PLC
- 3.4. Interfaces de Entrada/Salida y Específicas

Tema 4. Programación del PLC

- 4.1. Identificación de variables y asignación de direcciones
- 4.2. Lenguajes de programación
- 4.3. Programación de bloques funcionales
- 4.4. Entornos de programación
- 4.5. El estándar IEC 61131-3

Tema 5. Comunicaciones en los PLC

- 5.1. Enlaces estándar en el nivel físico
- 5.2. Redes de comunicación industriales
- 5.3. Ejemplos de Redes comerciales: Modbus, Profibus, ASI, CAN, HART, EtherCAT, etc.

Tema 6. Ejemplos de Aplicaciones de Automatización

- 6.1. Telecontrol del Ciclo Integral de Agua (Ciudad de Granada)
- 6.2. Control y Supervisión de una Estación de Esquí (Sierra Nevada)
- 6.3. Automatización de una fábrica de Papel (Torraspapel)

TEMARIO PRÁCTICO:

Seminario práctico 1: Utilización del editor de programa (STEP7) de los PLC del laboratorio

Seminario práctico 2: Programación y simulación de los PLC del laboratorio (Siemens)

Seminario práctico 3: Uso de la herramienta gráfica para diseñar un GRAFCET

Prácticas 1: Implementación guiada de una práctica para el control de una maqueta pequeña (maqueta elevador)

Prácticas 2: Programación de un automatismo sencillo (maqueta parking coches) con cualquiera de los lenguajes convencionales del estándar IEC-61131

Prácticas 3: Programación en GRAFCET de un proceso secuencial (maqueta almacenamiento y distribución de bloques)

Prácticas 4: Interconexión entre un PLC y un SCADA mediante el estándar OP

Como se puede extraer de los seminarios y prácticas planteadas la idea es que el alumno pueda realizar el ciclo completo de desarrollo de sistemas de control industriales, representado gráficamente en la figura 5, tal como se hace en la industria en general. Es decir, que disponga de un proceso que automatizar conectado con un PLC real que se pueda programar con una herramienta profesional lo más completa posible, y que a su vez el PLC esté comunicado mediante el estándar OPC con un SCADA comercial para poder supervisar y controlar el proceso real. En este escenario, el software se puede utilizar sin problemas en las prácticas (ya que hay versiones demos y licencias para estudiantes de dichas herramientas), pero será difícil tener un PLC para cada alumno (por el elevado coste que esto supondría) y no digamos un proceso real. Por ello, y para que el alumno pueda realizar unas prácticas siguiendo este esquema, hemos ideado una metodología e implementado unas herramientas (que se describen en la siguiente sección) que nos van a ayudar a conseguirlo.

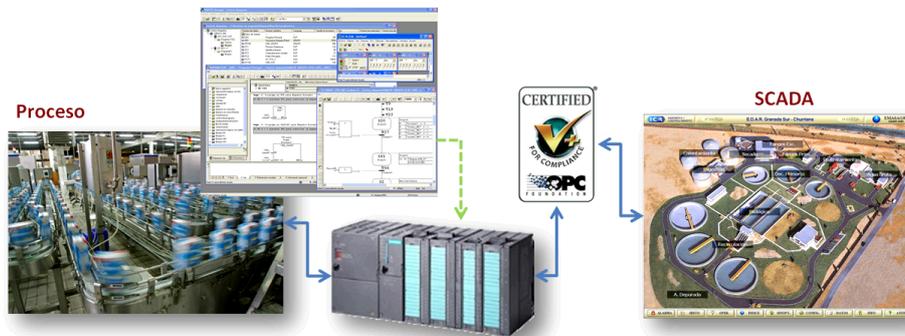


Figura 5. Ciclo completo de desarrollo de sistemas de control industriales

4 Equipos, herramientas y metodología a utilizar

Antes de describir los equipos, herramientas y metodología que se propone para esta asignatura, indicar que todo ello se ha adquirido y desarrollado bajo el paraguas de varios Proyectos de Innovación Docentes de la Universidad de Granada, coordinados por el Profesor Miguel Damas, y titulados:

1. Plataforma Docente Web para el Control de Maquetas de Simulación de Procesos Industriales (2007), galardonado con una mención honorífica en la convocatoria de Premios a la Innovación Docente 2008.
2. Laboratorio virtual para el ciclo completo de desarrollo de sistemas de control industriales (2011).
3. Nuevas metodologías de aprendizaje aplicadas a los sistemas de robótica industrial (2012).

Además, estas herramientas y metodología también han sido en parte publicadas en diversos congresos de docencia, tanto nacionales como internacionales [6,7,8,9].

4.1 Equipos y Herramientas desarrolladas

Utilizar un material adecuado de prácticas que motive al alumno en su aprendizaje es un factor muy importante a tener en cuenta. Por ejemplo, en [10] utilizan robots de LEGO para el estudio de sistemas embebidos o en [11] utilizan una maqueta de tres depósitos para asimilar de forma práctica conceptos fundamentales en el control de procesos. En nuestro caso, hemos optado por maquetas que emulan procesos industriales reales, ya que consideramos que son idóneas para la formación de técnicos en control de procesos. Concretamente, se han elegido varias maquetas fabricadas por Staudinger (ver figura 6), disponibles actualmente en el laboratorio de Control de la ETS de Ingeniería Informática y de Telecomunicación de la Universidad de Granada (ETSIT).



Figura 6. Algunas de las maquetas disponibles en el laboratorio

El laboratorio dispone también de un PLC de Siemens (muy utilizado en la industria en general), concretamente uno de la serie S7-300 (gama media). Se ha seleccionado este controlador porque se trata de uno de los fabricantes con mayor cota de mercado actualmente, y porque también proporciona un simulador de sus PLC (S7-PLCSim) y un ActiveX (S7-ProSim) que nos va a permitir integrar este simulador con otras aplicaciones: Maquetas virtuales y Servidor OPC desarrollados en nuestro caso.

Maquetas virtuales

Son programas que permiten emular gráficamente el comportamiento de las maquetas reales (ver figura 7). Es decir, que en función de las variables de salida que genera el programa realizado por el alumno y que se ejecuta en el simulador del controlador (S7-PLCSim), la maqueta virtual genera las variables de entrada necesarias sobre el simulador del controlador, para que este se comporte como si realmente estuviera conectado a la maqueta real. Además, la maqueta virtual muestra gráficamente los resultados de la ejecución de los programas para que el alumno compruebe si su práctica está funcionando correctamente o no. Algunas de las características y funciones más interesantes que ofrecen dichas maquetas virtuales son:

- Representación gráfica y animada acorde con la maqueta real simulada.
- Fácil interconexión con el simulador del controlador.
- Parametrización de las variables de los sensores y actuadores de las maquetas.
- Configuración de distintas posiciones iniciales de partida para que los programas de control de los alumnos se comporten de forma adecuada.
- Detección automática de errores de funcionamiento, con notificación de mensajes explicativos para que el alumno pueda depurar sus prácticas. Es decir, un supervisor de acciones no permitidas genera mensajes de errores a los alumnos cuando intentan realizar una operación que pueda causar un daño a la maqueta real o que no esté permitida según el funcionamiento programado.
- Autoevaluación de las prácticas de los alumnos, con indicación de la puntuación final en función de las iteraciones y distintas situaciones iniciales realizadas con éxito.
- Almacenamiento y recuperación del estado actual de la ejecución: direccionamiento de variables, configuraciones iniciales definidas, parámetros de autoevaluación, incidencias y errores detectados, etc.

Destacar sobre estas maquetas virtuales la funcionalidad de autoevaluación de las prácticas de los alumnos, donde se puede configurar distintas situaciones de partida. Por ejemplo, para la maqueta elevador, se puede seleccionar que inicialmente el elevador o el contenedor puedan estar en cualquier posición, lo cual influirá en la nota final que se obtenga. También podemos indicar el número de iteraciones (ejecuciones del programa desde distintas posiciones de partida de la maqueta) que se van a realizar y el tiempo de cada iteración. Al final de la autoevaluación se muestra una ventana en la que se indica la puntuación final, el número de iteraciones erróneas, el número de iteraciones correctas y la ventana de incidencias para que podamos analizar el porqué de las iteraciones erróneas.

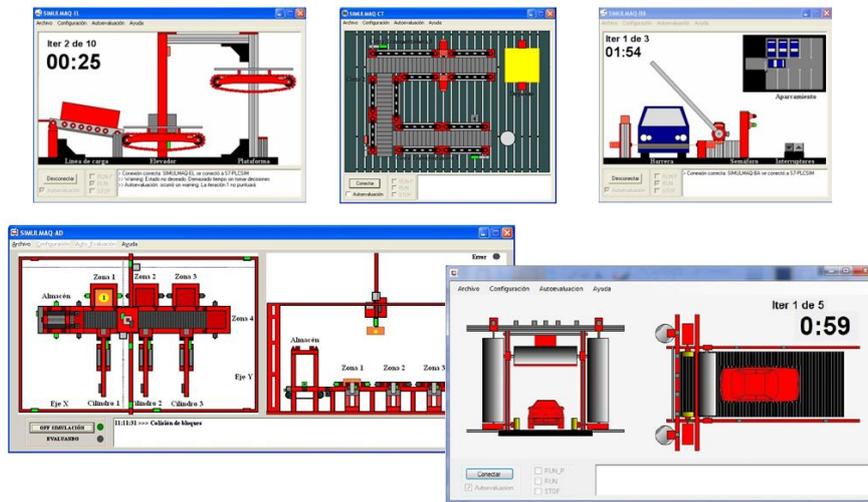


Figura 7. Maquetas virtuales desarrolladas para el laboratorio virtual: Maqueta elevador, maqueta cintas transportadoras, maqueta aparcamiento de coches, maqueta distribución y almacenamiento de bloques, y maqueta lava coches.

Servidor OPC-PLCSim

OPC (OLE/COM for Process Control) es un mecanismo de comunicación que permite la interconexión y el intercambio de datos entre dispositivos y aplicaciones software, soportado por la OPC Foundation, y que se ha convertido en el estándar para resolver la interoperabilidad en el entorno industrial. Por tanto, se ha desarrollado un servidor de comunicaciones según este estándar OPC que nos permite comunicarnos por un lado con el simulador de controladores de Siemens (S7-PLCSim) a través del ActiveX que proporciona el propio fabricante, y por otro lado con cualquier software de supervisión y control del mercado (SCADA) que implemente un cliente OPC.

4.2 Metodología

En cuanto a la metodología que se propone, como se ha indicado en la guía docente de la sección 3, se basa en una serie de seminarios y prácticas, acordes con el temario teórico y una serie de herramientas desarrolladas (descritas en la sección anterior), para que el alumno pueda abordar de forma práctica todas las fases necesarias a la hora de realizar un sistema completo de automatización industrial. Es decir, que el alumno pueda realizar prácticas no solo de la programación de los equipos de control sino también de la configuración del software de supervisión y control necesarios en las salas de control, así como también de la intercomunicación entre ambos, y siempre utilizando los equipos (PLCs), software (SCADA y Entornos de programación) y protocolos (OPC) que más se usan actualmente de forma comercial en la industria.

Las clases prácticas del curso comienzan con varios seminarios dedicados a introducir los entornos de programación y los lenguajes del estándar IEC 61131-3 [12] que se utilizan en la automatización industrial. Concretamente, se imparte un seminario sobre cómo se utiliza el editor de programa STEP7 de los PLC de Siemens, y otro sobre cómo se programan dichos PLC, y en especial de cómo se realiza un SFC (Sequential Function Chart).

Para completar los seminarios anteriores, a continuación en las prácticas 1 y 2 se realizan implementaciones guiadas por el profesor utilizando maquetas sencillas con el objetivo de ilustrar la utilización de elementos de programación habituales en los automatismos industriales (biestables, temporizadores y contadores), la ventaja de utilizar SFC, así como para explicar el manejo de las maquetas virtuales. Concretamente en esta fase se usan la maqueta elevador, la maqueta aparcamiento de coches, y la maqueta cintas transportadoras.

En la práctica 3 se le propone al alumno que implemente desde el principio un programa para el control de una maqueta más compleja, utilizando los lenguajes y herramientas ya introducidos en las fases anteriores. Para ello se utilizan las maquetas virtuales de distribución y almacenamiento de bloques y la de lava coches. Con dichas maquetas virtuales el alumno puede probar y depurar su programa cuantas veces quiera sin dañar las maquetas reales. Además, una vez tenga depurada su implementación, el alumno podrá lanzar una autoevaluación que le indicará automáticamente una puntuación final orientativa del grado de corrección del programa. Con esta práctica se pretende que el alumno afiance sus conocimientos de

programación de automatismos utilizando un entorno de programación comercial, así como los lenguajes del estándar 61131-3, y en especial, el SFC para implementar las tareas secuenciales de los procesos de fabricación industriales.

Finalmente, en la última práctica se completa el ciclo de desarrollo en la automatización de un proceso industrial (ver figura 8) introduciendo el estándar de comunicaciones OPC, y el software para la supervisión y control de procesos (SCADA). Para ello se utiliza un servidor OPC, implementado a propósito para este laboratorio virtual, que permite comunicar el simulador del PLC con el SCADA. En esta práctica, el alumno debe desarrollar una aplicación de supervisión y control para cualquiera de las maquetas disponibles, utilizando una herramienta comercial (como por ejemplo InTouch de Wonderware) y un servidor OPC-PLCSim para acceder a las variables del PLC simulado, de la misma forma que se haría en una aplicación real.

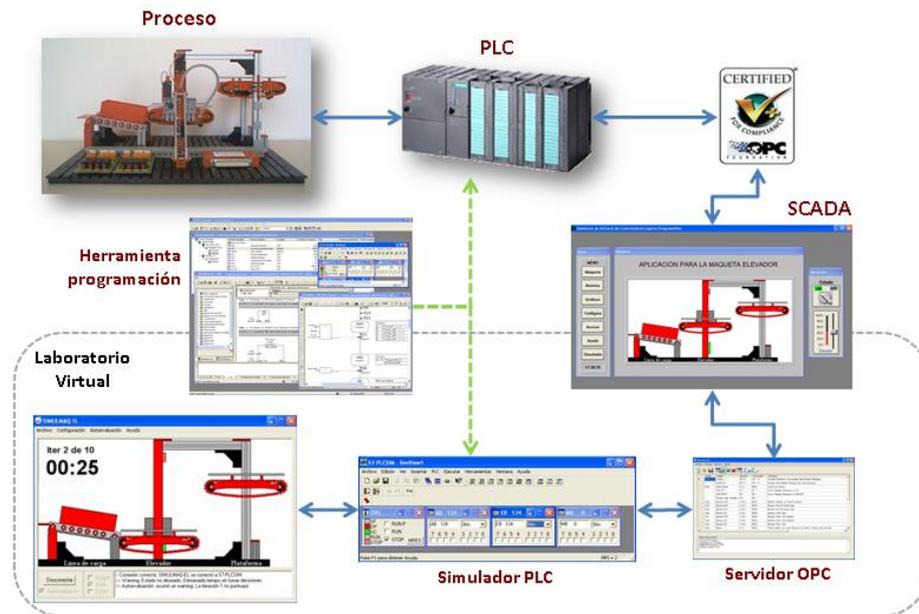


Figura 8. Esquema general de trabajo con el laboratorio virtual propuesto

5 Conclusiones

En este trabajo en primer lugar se ha presentado la asignatura Controladores Lógicos Programables como optativa del Grado de Ingeniería Informática de la Universidad de Granada, justificando su impartición en esta titulación dentro de la materia de complementos de Informática Industrial. A continuación, se han presentado tanto la metodología como los equipos y herramientas que se van a utilizar en dicha asignatura, en torno a un laboratorio virtual desarrollado para ello.

El facilitar un laboratorio virtual donde se pueda trabajar como si se estuviera en un entorno industrial real es fundamental para poder evaluar adecuadamente que los alumnos hayan adquirido una competencia importante para ellos.

La metodología propuesta se ha presentado tanto desde el punto de vista técnico como educacional. Concretamente, se han mostrado los componentes tanto hardware como software que conforman el sistema, y se han descrito algunas de las maquetas con las que se trabajan. También, y de forma muy resumida, se ha presentado el escenario pedagógico que se sigue, donde se utilizan adecuadamente las herramientas presentadas en este trabajo.

La posibilidad de realizar prácticas en las distintas fases de realización de un proyecto (programación de controladores, interfaces de comunicaciones y software de supervisión y control) hace que el alumno tenga una visión completa del ciclo de desarrollo que todo ingeniero ha de realizar en la automatización de cualquier proceso industrial.

Las herramientas presentadas permiten que el alumno pueda organizarse mejor y eliminar barreras físicas (desplazamientos) a la hora de realizar las prácticas de asignaturas que hasta ahora necesitaban una presencia forzosa por la utilización de equipos específicos. También se minimiza el número de pruebas sobre las maquetas reales evitando daños y pérdidas de tiempo tanto del alumno como del profesor que ha de evaluar las prácticas.

Se debe de destacar también que la función de auto evaluación que incorpora la herramienta permite de forma automática realizar un chequeo exhaustivo de las prácticas proporcionando al final una calificación orientativa al alumno.

Luego, el esquema de trabajo presentado para la asignatura Controladores Lógicos Programables tiene, a nuestro parecer, un gran potencial didáctico, ya que permite que el alumno pueda trabajar con herramientas de programación y equipos reales que se emplean habitualmente en el control de cualquier proceso industrial, sin que su disponibilidad, coste, tamaño, o ubicación sean un problema.

Agradecimientos. Al Secretariado de Innovación Docente de la Universidad de Granada por los Proyectos de Innovación Docente concedidos que han posibilitado la adquisición de los equipos necesarios para desarrollar la metodología docente presentada en este trabajo.

Referencias

1. M. Damas, J. Ortega, O. Baños: “Una iniciativa para la coordinación y difusión de la docencia del perfil de Ingeniería de Computadores del Grado de Informática de la UGR”. Enseñanza y Aprendizaje de Ingeniería de Computadores. Revista de Experiencias Docentes en Ingeniería de Computadores. ISSN: 2173-8688, Nº2, pp.73-80, 2012.
2. J.A. Rehg, G.J. Sartori, “Programmable Logic Controllers”, ISBN: 978-0135048818, Prentice Hall, 2 edition, 2008.
3. S.A. Boyer, “SCADA: Supervisory Control and Data Acquisition”, ISBN: 978-1556178771, ISA. 3 edition, 2004.

4. W. Mahnke, S. Leitner, M. Damm, "OPC Unified Architecture", ISBN: 978-3540688983, Springer, 1 edition, 2009.
5. H.Meyer: "Manufacturing Execution Systems: Optimal Design, Planning, and Deployment". ISBN: 9780071623834, McGraw-Hill Companies, Inc., 2009.
6. M.Damas, H.Pomares, I.Rojas: "A new E-learning Strategy for Industrial Computation Students". IASK International Conference Teaching and Learning, Aveiro (Portugal), 2008.
7. M.Damas, H.Pomares, I.Rojas: "Use of an E-Learning Platform to Create a Virtual Lab in Computer and Electrical Engineering Courses". Improving University Teaching IUT'08, Transforming Higher Education Teaching and Learning in the 21st Century, Glasgow (United Kingdom), 2008.
8. M.Damas, G.Olivares, H.Pomares, A.Olivares: "Laboratorio Virtual basado en Web para el ciclo completo de desarrollo de sistemas de control industriales". Workshop en Informática Industrial (WIIND), ISBN: 84-692-2381-9, Universidad Rey Juan Carlos, Madrid, 2009.
9. M.Damas, H.Pomares, G.Olivares: "Innovación en la docencia práctica de asignaturas relacionadas con la informática industrial". 1ª Jornadas Andaluzas de Innovación Docente Universitaria. ISBN: 978-84-692-7263-3, Córdoba, 2009.
10. S. H. Kim, J. W. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms", IEEE Transactions on Education, Vol. 52, no. 1, pp. 99-108, 2009.
11. R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S. Dormido, "Development of a Web-Based Control Laboratory for Automation Technicians: The Three-Tank System", IEEE Transactions on Education, Vol. 51, no. 1, pp.35-44, 2008.
12. J. Karl-Heinz, M. Tiegelkamp, "IEC 61131-3: Programming Industrial Automation Systems", ISBN: 978-3540677529, Springer, 1 edition, 2001.
13. C. Buiu, "Hybrid Educational Strategy for a Laboratory Course on Cognitive Robotics", IEEE Transactions on Education, Vol. 51, no. 1, pp.100-107, 2008.
14. M. Duarte, B. P. Butz, S. M. Miller, A. Mahalingam, "An Intelligent Universal Virtual Laboratory (UVL)", IEEE Transactions on Education, Vol. 51, no. 1, pp. 2-9, 2008.
15. B. Barros, T. Read, M. F. Verdejo, "Virtual Collaborative Experimentation: An Approach Combining Remote and Local Labs", IEEE Transactions on Education, Vol. 51, no. 2, pp.242-250, 2008.
16. Z. Nedic, J. Machotka, A. Nafalski, "Remote laboratories versus virtual and real laboratories," in Proc. 33rd Frontiers in Education Conf., Boulder, CO, pp. T3E.1–T3E.6, 2003.

La asignatura Informática Industrial y la función del Ingeniero Informático en la Industria

G. Olivares, M. Damas, F. Gómez

Departamento de Arquitectura y Tecnología de Computadores. ETSI Informática y de Telecomunicación. Universidad de Granada
Granada, España
{gonzalo, mdamas, frgomez}@ugr.es

Resumen. Describimos en este trabajo los objetivos y el desarrollo previsto para la asignatura “Informática Industrial” que se impartirá como optativa en el grado en Ingeniería Informática de la Universidad de Granada, a partir del curso 2013-14. En el artículo se analiza cómo se organiza la docencia teórica y práctica con el fin último de dar a conocer el importante papel del Ingeniero Informático actual en el mundo de la Ingeniería de Control de Procesos Industriales y reflexionar cómo se puede motivar a los alumnos en una disciplina que, sin olvidar los fundamentos de los sistemas de control, debe dirigirse a conseguir que el futuro Ingeniero Informático sepa cuáles van a ser las funciones y competencias que se esperan de un profesional y los conocimientos globales que necesitará para comunicarse con los Ingenieros de Procesos, a la hora de diseñar e implantar Sistemas de Información, Supervisión y Control para la Industria.

Palabras Clave: Informática Industrial, Control, Ingeniería Informática, motivación del alumnado.

Abstract. In this paper we describe the “Industrial Computing” subject that is taught as an elective subject in Computer Engineering degree at the University of Granada. We analyze how the theoretical and practical teaching is organized in order to raise awareness of the important role of the current Computer Engineers in the Industrial Control world and how to motivate students in a discipline that, without forgetting the fundamentals of control systems, must be addressed to ensure that the future engineer knows what will be the functions roles and responsibilities expected of a professional and comprehensive knowledge they need to communicate with engineers industrial process to implement Information, Monitoring and Control Systems for Industry.

Keywords: Industrial Computing, Control, Computer Engineering, student motivation.

1 Introducción

La nueva asignatura “*Informática Industrial*” del grado de Ingeniería Informática de la Universidad de Granada, es una disciplina que se impartirá como optativa a partir del curso 2013-2014 y ha sido planificada basándonos en la experiencia acumulada tras impartir durante varios años las asignaturas “*Informática Industrial*” de la anterior Ingeniería Técnica Informática y “*Sistemas de Control Digital*”, asignatura de Ingeniería Informática en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la misma universidad, así como nuestra propia experiencia profesional y empresarial en el ámbito del Control Industrial en las últimas dos décadas. Esta asignatura está íntimamente relacionada, y se complementa adecuadamente, con otra asignatura optativa: “*Controladores Lógicos Programables*” que se imparte en el mismo curso.

El alumno del Grado en Ingeniería Informática, de partida, piensa que históricamente, e incluso en la actualidad, los únicos ingenieros que trabajan en la automatización y control de procesos son casi exclusivamente Ingenieros Industriales o bien Ingenieros en Electrónica Industrial. Algunas de las funciones del Ingeniero de Control están relacionadas con el diseño, desarrollo e implantación de la instrumentación de control (sensores, actuadores, reguladores,..), con la programación de Controladores Lógicos Programables, con el diseño y configuración de redes locales industriales y buses de campo y también con el desarrollo de las aplicaciones informáticas de monitorización y control que se llevan a cabo en Centros de Control. Todas ellas son funciones que perfectamente puede llevar a cabo el profesional informático con los complementos de formación apropiados.

2 Motivación

No es fácil que el alumno deje de pensar que este mundo le es ajeno; casi siempre pensará que su papel profesional potencial se sitúa más bien dentro del equipo de Informáticos de Gestión, trabajando justo en la parte superior de la pirámide CIM (*Computer Integrated Manufacturing*). Nuestro primer objetivo es hacerle ver que el papel del profesional informático dentro de este sector es esencial, aunque eso sí, tendrá que convivir y entenderse con otros profesionales de otras ramas de la Ingeniería, y tendrá que hablar con propiedad el mismo lenguaje que ellos. Para lograr convencerlo disponemos de un curso de 6 créditos.

Si, según el diccionario de la Real Academia de la Lengua, “la Automática es la disciplina que trata de métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la ejecución de una tarea física y mental previamente programada”, este concepto es actualmente mucho más amplio, ya que, si hablamos de la Automática aplicada a la Industria, aparte de hacer énfasis en la sustitución del operador, habría que incluir también otras muchas funciones, tales como: la mejora de la calidad de los productos manufacturados y de las prestaciones de los sistemas a controlar, el desarrollo de métodos de fabricación flexible de acuerdo a las condiciones variantes del mercado, la seguridad global de personas y bienes de producción, el ahorro de costes de materia prima, la supervisión

en forma jerárquica de todos los procesos y la sincronización de las diferentes áreas de los procesos productivos, desde el nivel de campo hasta el nivel de gestión. El Ingeniero Informático debe jugar aquí un papel esencial, sobre todo en el diseño y programación de los sistemas de control local y distribuido, en la implantación de los sistemas de comunicaciones (buses de campo y redes locales industriales), en la implementación de sistemas de control y supervisión, en el despliegue de servidores y bases de datos, en el diseño de los sistemas SCADA, o en la integración con sistemas ERP.

En la mayoría de las Escuelas de Ingeniería Industrial la asignatura “Informática Industrial” se dirige hacia enfoques de programación, o de Ingeniería de Software, en general. En nuestro caso, aunque la asignatura tenga el mismo nombre, el objetivo es justamente el contrario: se trata de acercar al futuro Graduado en Informática al mundo Industrial, intentando conseguir que pierda el miedo a trabajar en el sector, y ampliando sus horizontes profesionales, haciéndole ver que su vocación se puede abrir a otros ámbitos, más allá de la programación de software de gestión, del desarrollo de aplicaciones web, del mantenimiento de redes y servidores o de la implementación de “apps” para teléfonos y tablets, entre otros.

2.1 Objetivos de aprendizaje

En términos generales, como hemos dicho anteriormente, el Ingeniero Informático Industrial tendrá que convivir y formar parte de equipos multidisciplinares (Instrumentistas, Ingenieros de Campo, Ingenieros de Procesos, Ingenieros Eléctricos, o Ingenieros de Producción). Así, los objetivos específicos de aprendizaje son:

- Introducir al alumno en los conceptos básicos sobre sistemas de regulación y control de sistemas lineales.
- Conocer la evolución histórica de la Informática Industrial.
- Aprender a modelar sistemas y procesos, utilizando herramientas software de análisis y simulación.
- Conocer los fundamentos de controladores PID, y aprender a ajustar experimentalmente lazos de regulación.
- Conocer los fundamentos de la arquitectura y de los lenguajes de programación normalizados de Controladores Lógicos Programables.
- Diseñar sistemas de adquisición de datos, supervisión y control.
- Aprender a evaluar los principales sensores, actuadores, sistemas de control, ordenadores industriales y terminales remotas que habitualmente se utilizan en la Industria.
- Conocer los principales buses de campo y redes locales industriales, así como las configuraciones de control distribuido.
- Diseñar y configurar sistemas SCADA para la supervisión y control de plantas industriales.
- Diseñar los elementos de centros de control y arquitecturas de supervisión distribuida
- Tener contacto con aplicaciones reales de automatización industrial.

3 Contenidos de la asignatura

3.1 Programa de Teoría

Tras la presentación de la asignatura, se presenta un tema de “Introducción a los Sistemas de Control” destinado a dar al alumno una visión rápida de la materia que se va a estudiar a lo largo del curso, de una manera global y resumida, al objeto de conseguir que el alumno adquiera rápidamente los conocimientos necesarios para iniciar las prácticas experimentales de laboratorio. En este tema se presentan los conceptos de sistemas de regulación en lazo abierto y lazo cerrado, control secuencial y se da una visión del control por computador. A continuación se analizan con algo más de detalle los elementos de un lazo de control: medidores, actuadores, y controladores.

Para finalizar el tema de introducción, se presentan varios ejemplos de sistemas de control reales. Finalizado este tema, el alumno tiene que realizar y presentar un conjunto de ejercicios que le obligarán a hacer una reflexión para identificar los elementos y la arquitectura de diversos sistemas de control mediante el análisis de problemas concretos (*regulación de insulina para enfermos diabéticos, control automático de la velocidad de un coche, control del timón de un barco a motor o a vela, identificación de elementos de un vehículo “Segway”, estudio de un sistema de control de la climatización del habitáculo de un vehículo, etc ..*).

A continuación se introduce al alumno en la teoría de control clásica, estudiando el modelado de sistemas lineales mediante funciones de transferencia y ecuaciones de estado, para posteriormente estudiar la respuesta transitoria de sistemas de primer y segundo orden, y la determinación de estabilidad en sistemas de control continuo. Además se inicia al alumno en el diseño de controladores. En todos los casos, se presentan ejemplos realizados con Matlab y Simulink, para que el alumno entienda mejor los fundamentos teóricos a partir de aplicaciones prácticas. Así, recurrimos al modelado de sistemas de control sencillos pero ilustrativos, tales como a) el control de velocidad de crucero de un coche, b) el análisis del comportamiento térmico de una vivienda, como parte de un sistema domótico, c) la regulación del nivel de un tanque de líquido, y d) el modelado de un péndulo invertido, como fundamento físico del vehículo Segway. La Figura 1, ilustra el modelado de dichos sistemas.

Pasamos a continuación, según el programa diseñado, al estudio de un tema dedicado a “Sensores y Actuadores”, desglosado de la siguiente forma:

Fundamentos y clasificación de sensores. Características estáticas y dinámicas. Criterios de selección. Tipos normalizados de señales de salida. Galgas extensiométricas, presión y células de carga. Temperatura (PTC, NTC, Termistores, pirómetros de radiación,..). Posición lineal y angular: codificadores incrementales y absolutos. Nivel y Caudal. Relés, contactores, solenoides. Motores: continua, alterna, de pasos. Actuadores electrohidráulicos y electro-neumáticos. Válvulas. Actuadores térmicos.

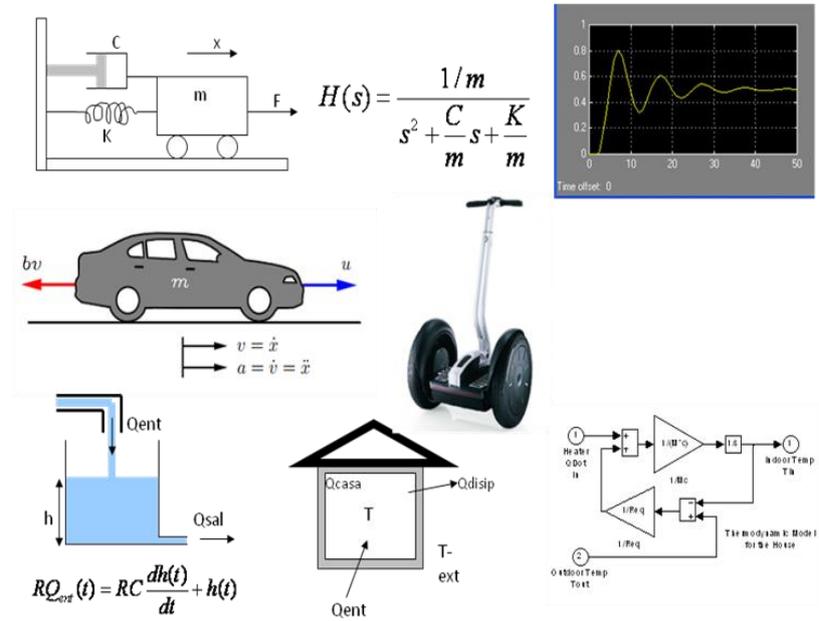


Figura 1. Ejemplos de modelado de sistemas lineales.



Figura 2. Ejemplos de sensores y actuadores industriales

Además de estudiar los fundamentos de los principales sensores, se pretende conseguir que el alumno pueda identificar los distintos parámetros que le ayudarán a seleccionar adecuadamente la instrumentación que necesitaría para la implementación de sistemas de control (ver figura 2). En todos los casos, una vez revisados los fundamentos de los medidores o actuadores, les presentaremos

catálogos y documentos de productos de distintos fabricantes para que puedan analizar aspectos tales como: rango de medida, resolución, precisión, ancho de banda, histéresis, condiciones de sobre-rango, requisitos de instalación y mantenimiento, condiciones ambientales de medida, etc.

Más adelante estudiamos con más profundidad las distintas opciones de controladores, según los siguientes apartados:

Fundamentos de control. Controladores "todo-nada" . Controladores continuos. Medida de la calidad de un sistema de control. Métodos de ajuste. Control en cascada. Controladores analógicos. Control secuencial. Control mediante lógica cableada. Controladores Lógicos programables.

Se revisan los principales métodos de ajuste experimental de lazos de regulación y los parámetros que nos permiten diseñar una respuesta transitoria adecuada. Se estudian las características de los reguladores industriales comerciales, y los fundamentos del control secuencial. Tras visualizar algunos vídeos de procesos industriales de control secuencial, se plantean algunos ejemplos de diseño combinatorial, para a continuación, transmitirle al alumno la necesidad de utilizar Controladores Lógicos Programables-PLCs. Dado que existe otra asignatura optativa específica en la titulación, se dedica exclusivamente un pequeño apartado del tema a esta materia, dirigido sólo a aquellos alumnos que por cualquier circunstancia no la han elegido. Se estudia de manera global y esquemática la arquitectura y ciclos de funcionamiento de los PLCs, se citan los lenguajes de programación normalizados (IEC 1131-C) y se presentan algunos sencillos ejemplos de programación en SFC, a partir de las especificaciones de diseño.

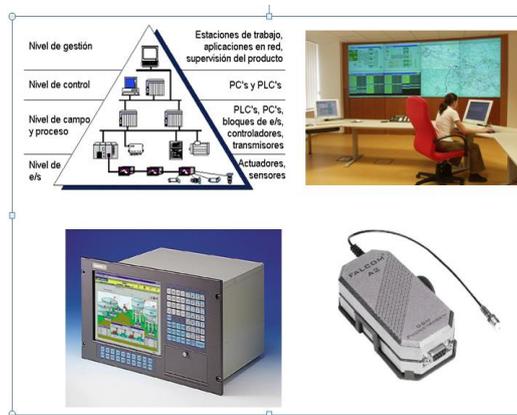


Figura 3. Arquitecturas, sistemas de control, redes y equipos de comunicaciones.

A continuación, una vez asimilados los conocimientos básicos, enfocamos el resto de la asignatura a revisar las principales Arquitecturas, Redes de Control y Sistemas de Comunicaciones Industriales según el siguiente desglose:

Sistemas en modo supervisión. Control Digital Directo. Control Distribuido. Concepto CIM. Ordenadores Industriales y Periféricos. Centros de Control. Buses de Campo y Redes Locales Industriales.

Actualmente, son muchos los Ingenieros Informáticos que trabajan en el desarrollo de aplicaciones de supervisión, control y adquisición de datos (SCADA–“Supervisory Control and Data Acquisition”). Le dedicaremos un apartado específico al estudio de las principales características de este tipo de herramientas de software configurable: creación de esquemas sinópticos, tags, históricos, informes, alarmas, interfaces de comunicaciones vía OPC, etc. Se muestran diversos ejemplos de aplicaciones reales de redes de supervisión de sistemas de abastecimiento de agua, plantas de fabricación de papel, sistemas de bombeo, control integral de una estación de esquí, y control de instalaciones de edificios.

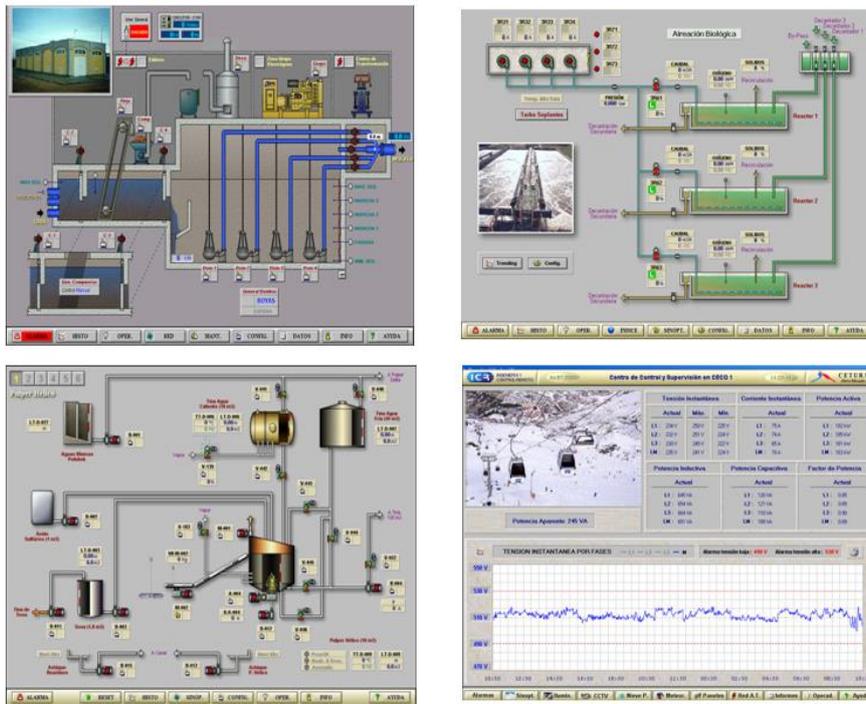


Figura 4. Ejemplos de supervisión de procesos mediante software SCADA

3.2 Programa de prácticas

Como siempre ocurre en asignaturas con una fuerte componente experimental, consideramos también que las prácticas de Informática Industrial son esenciales para la correcta formación del estudiante; incluso, se podría asegurar que gran parte de la materia teórica está dirigida a adquirir los conocimientos necesarios para realizar las

prácticas, como objetivo didáctico último. El material que disponemos en el laboratorio requiere el desarrollo de varios seminarios específicos (no impartidos previamente en las clases teóricas) para que el alumno conozca en profundidad el equipamiento que va a utilizar.

Disponemos de los siguientes equipos físicos:

- Tarjetas *Sensonic* que incorpora dispositivos sensores y actuadores, sobre la cual el estudiante tendrá que realizar varios sistemas de control.
- Maqueta hidráulica.
- Maqueta de motores.
- Maqueta térmica.
- Conjunto de módulos conectados vía bus de campo, del fabricante ICP-DAS.

Por otro lado, disponemos de la herramienta de software SCADA Intouch del fabricante Wonderware, posiblemente el SCADA más utilizado en el mundo de control industrial. La tarjeta *Sensonic* se ha diseñado íntegramente entre el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada y la empresa *Seven-Solutions*, spin-off fundada también por profesores del mismo departamento.

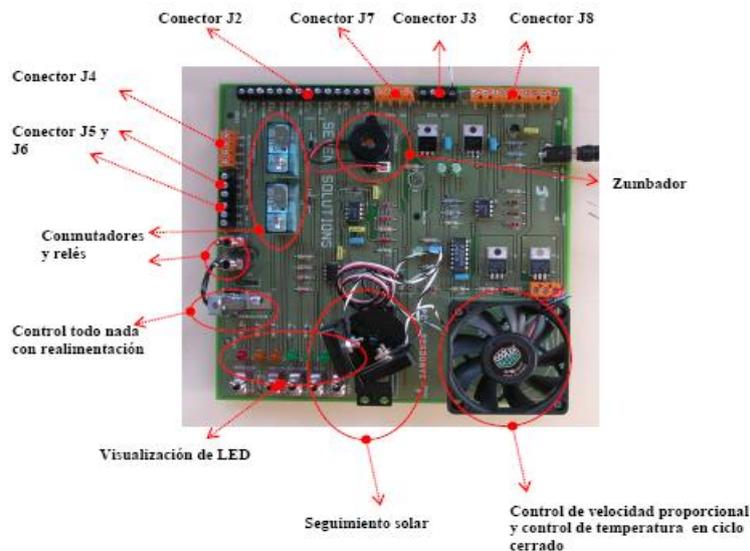


Figura 5. Fotografía de la tarjeta Sensonic

Con los elementos sensores incorporados, el estudiante puede realizar medidas continuas de temperatura, velocidad de giro de un motor, e intensidad luminosa en placas fotovoltaicas, además de detección de presencia mediante efecto Hall, y lectura de la posición de interruptores. Como elementos actuadores dispone de servomotor de corriente continua, solenoide lineal, zumbador, relés de potencia y calefactor de regulación continua. (Ver Figura 5). La tarjeta es de muy bajo coste y disponemos de 25 unidades para que el alumno pueda trabajar individualmente realizando

controlador de tipo todo-nada, controladores PID discretos de temperatura, control digital de posición, y actuaciones de control secuencial.

Los sensores y actuadores de la tarjeta Sensonic se supervisan y controlan mediante un conjunto de módulos de la serie 7000 del fabricante ICP-DAS, conectados mediante un bus de campo propio sobre interfaz físico RS-485.

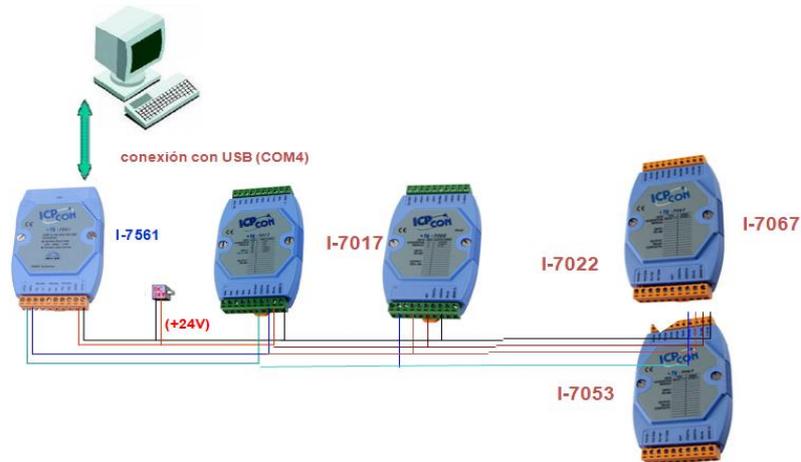


Figura 6. Módulos ICP-DAS

Los módulos de conversión A/D, D/A y de entradas y salidas digitales (ver Figura 6), se conectan vía USB con el ordenador que manejará el alumno para realizar el software de supervisión y control de la tarjeta Sensonic y de las restantes maquetas. Para ello utilizará el SCADA Intouch Wonderware, sin coste de licencias, siempre que no se utilicen más de 32 variables (tags) de entrada/salida.

Antes de empezar a controlar la tarjeta Sensonic, el alumno recibe un seminario didáctico de Intouch básico, para a continuación realizar una simulación libre de un proceso industrial, al objeto de familiarizarse con esta herramienta de configuración, que le permite realizar fácilmente esquemas sinópticos de procesos, gráficos de tendencias en tiempo real, históricos, alarmas, etc. (ver Figura 7).

A continuación se imparte un seminario sobre OPC, para posteriormente utilizar el servidor OPC de IPC-DAS y OPCLink de Intouch. Una vez configurado el sistema, y creados los *tags* (variables de E/S), el objetivo siguiente es crear una aplicación SCADA para medir las magnitudes de todos los sensores y actuar sobre los elementos de control disponibles en la tarjeta Sensonic, tanto discretos como continuos.

Más adelante se plantean varios objetivos: a) control de temperatura todo-nada con histéresis, b) control PID de temperatura con algoritmo recursivo, c) ajuste experimental de constantes PID en función de la respuesta transitoria, d) control de servomecanismo y e) posicionamiento de mini-paneles fotovoltaicos para conseguir la máxima intensidad luminosa.

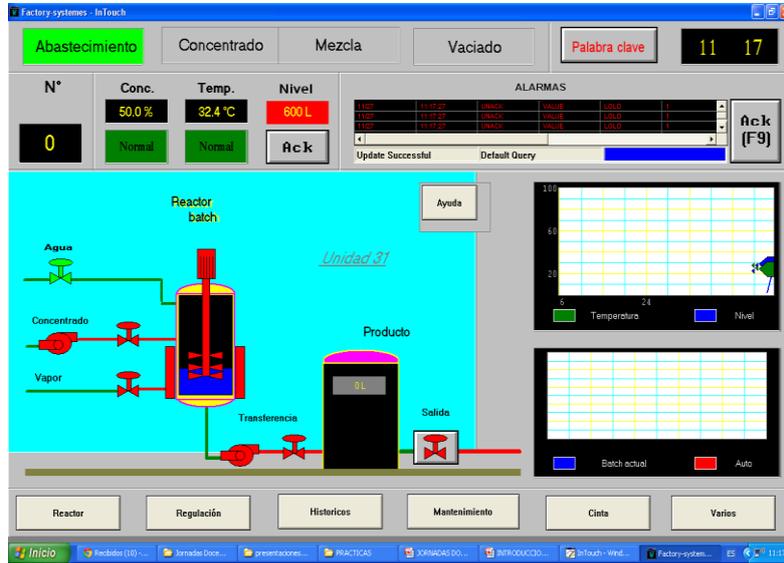


Figura 7. Pantalla ejemplo realizada con SCADA Intouch

Más adelante se plantean varios objetivos: a) control de temperatura todo-nada con histéresis, b) control PID de temperatura con algoritmo recursivo, c) ajuste experimental de constantes PID en función de la respuesta transitoria, d) control de servomecanismo y e) posicionamiento de mini-paneles fotovoltaicos para conseguir la máxima intensidad luminosa.

Disponemos también de tres maquetas (hidráulica, térmica y de motores) para prácticas opcionales y de demostración (ver Figura 8).

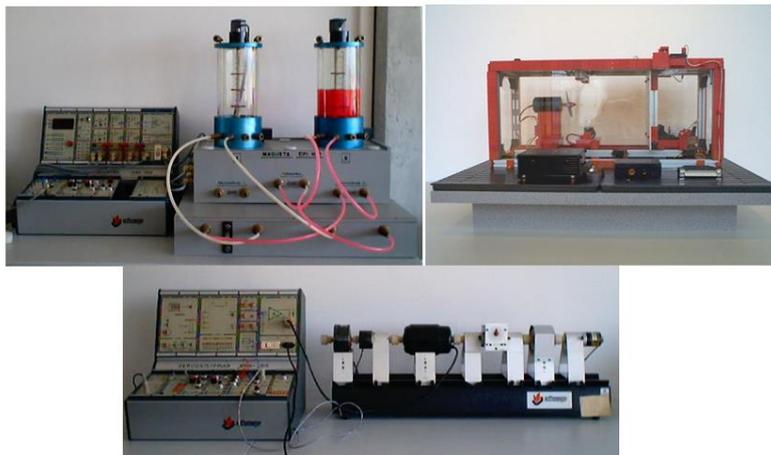


Figura 8. Maquetas hidráulica, térmica y de control de motores.

4 Conclusiones

Se ha presentado en este artículo los objetivos y contenidos de la asignatura optativa “Informática Industrial” del Grado en Informática que se imparte en la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada. Esta asignatura, eminentemente práctica pretende transmitir al alumno los conocimientos suficientes para que el futuro profesional informático pueda integrarse en equipos multidisciplinares de empresas de Ingeniería de Control Industrial. Por experiencia en asignaturas precedentes, anteriores al plan de estudios actual, la mayoría de los alumnos tiene un gran interés, y tras las primeras clases se consigue un alto grado de motivación, que se incrementa cuando entra en el laboratorio y ve que con bastante facilidad realiza las prácticas satisfactoriamente. El estudiante agradece que, aunque los procesos a controlar se desarrollan sobre maquetas, las herramientas que utilizan son las que realmente se van a encontrar en el ámbito real, si finalmente desarrollan su vida profesional en el campo industrial.

Agradecimientos. Al Secretariado de Innovación Docente de la Universidad de Granada por los Proyectos de Innovación Docente concedidos que nos han permitido financiar la adquisición de algunas de las maquetas de laboratorio utilizadas en prácticas.

Referencias

1. M. Damas, J. Ortega, O. Baños: “Una iniciativa para la coordinación y difusión de la docencia del perfil de Ingeniería de Computadores del Grado de Informática de la UGR”. Enseñanza y Aprendizaje de Ingeniería de Computadores. Revista de Experiencias Docentes en Ingeniería de Computadores. ISSN: 2173-8688, N°2, pp.73-80, 2012.
2. S.A. Boyer, “SCADA: Supervisory Control and Data Acquisition”, ISBN: 978-1556178771, ISA. 3 edition, 2004.
3. W. Mahnke, S. Leitner, M. Damm, “OPC Unified Architecture”, ISBN: 978-3540688983, Springer, 1 edition, 2009.
4. M.Damas, H.Pomares, I.Rojas: “A new E-learning Strategy for Industrial Computation Students”. IASK International Conference Teaching and Learning, Aveiro (Portugal), 2008.
5. M.Damas, G.Olivares, H.Pomares, A.Olivares: “Laboratorio Virtual basado en Web para el ciclo completo de desarrollo de sistemas de control industriales”. Workshop en Informática Industrial (WIIND), ISBN: 84-692-2381-9, Universidad Rey Juan Carlos, Madrid, 2009.
6. M.Damas, H.Pomares, G.Olivares: “Innovación en la docencia práctica de asignaturas relacionadas con la informática industrial”. 1ª Jornadas Andaluzas de Innovación Docente Universitaria. ISBN: 978-84-692-7263-3, Córdoba, 2009.

Servidores Web de Altas Prestaciones

Pedro A. Castillo, Jose L. Bernier, Antonio M. Mora,
Pablo García-Sánchez, M. G. Arenas

Departamento de Arquitectura y Tecnología de Computadores.
Universidad de Granada
Granada, España
pedro@atc.ugr.es

Resumen. Para crear un sitio web exitoso debemos ser cuidadosos con muchos detalles, ya que la arquitectura, la selección del hardware, o la implementación de las aplicaciones afectarán al rendimiento del sistema completo. Es más, un sitio que no esté preparado para dar servicio a un número creciente de clientes o usuarios, a la larga puede llevar al fracaso a la empresa. En este trabajo, se presenta una visión general de la nueva asignatura Servidores Web de Altas Prestaciones, del nuevo plan de estudios del Grado en Ingeniería Informática de la Universidad de Granada, así como la metodología a seguir para impartir tanto la teoría como las prácticas. A través de un aprendizaje principalmente práctico tratamos de que el alumno descubra cómo diseñar, tanto a nivel de hardware de red como de configuración de software, un servidor web de altas prestaciones. A lo largo de la asignatura pretendemos dar una visión detallada de los conceptos y herramientas utilizadas para crear o mantener la infraestructura de un sistema de forma adecuada. Estudiaremos las ideas y prácticas para asegurar la escalabilidad del sistema, asegurando el servicio a nuevos usuarios y manteniendo el sistema activo.

Palabras Clave: servidor web, disponibilidad, balanceo de carga, docencia

Abstract. To create a successful website, we must be careful with many details, as the architecture, hardware selection, and implementation of applications, and how they affect the performance of the entire system. Moreover, a site that is not prepared to serve a growing number of customers or users, can eventually lead to failure of the company. In this paper, we present an overview of the new subject "High Performance Web Servers" in the new curriculum's Degree in Computer Engineering from the University of Granada, as well as the methodology to teach both theory and practice. Through a practical learning, we try to make the student learn how to design both the network hardware and software configuration of a high performance web server. Throughout the course we intend to give a detailed overview of the concepts and tools used to create and maintain the infrastructure of a system properly. We will study the ideas and practices to ensure the scalability of the system and the service to new users.

Keywords: web server, availability, load balancing, teaching

1 Introducción

Al crear un sitio web, los administradores de sistemas tienen por delante un duro trabajo inicial para hacer más fácil el mantenimiento posterior.

Como primera y más fácil aproximación, lo habitual es montar un servidor en una sola máquina. Esto, que resulta fácil y rápido, a la larga necesitará mucho más trabajo de mantenimiento, y probablemente haga que el sitio web experimente graves problemas. Las tareas derivadas de los problemas de hardware y software, así como dar servicio a diferentes cargas de trabajo, acaban siendo un trabajo demasiado costoso en servidores sencillos que deben atender gran tráfico [1].

La única solución a estos problemas pasa por transformar el sistema en una granja web (web farm) compuesta de varios servidores dedicados cada uno a una tarea concreta. Una granja web es un sistema escalable y con alta disponibilidad. Puede resultar mucho más complejo y trabajoso instalarlo y configurarlo, pero es el tipo de sistema que puede soportar un alto tráfico de red y dar servicio a millones de usuarios [2,5].

Del conjunto de servidores dedicados a servir contenido web a los usuarios finales, algunos pueden servir contenido HTML estático, mientras que otros pueden albergar las bases de datos, o bien ofrecer contenido dinámico. Se trata de sistemas complejos, pero muy flexibles a la hora de diseñarlos y construirlos.

A pesar de la importancia de estos temas, en gran parte de las titulaciones de Ingeniería Informática existía una carencia en cuanto a que ciertos temas referentes a tecnologías web no se estudiaban.

En octubre de 2010 comenzó la implantación de los estudios del nuevo Grado en Ingeniería Informática por la Universidad de Granada. Como consecuencia de la visión de las necesidades del mercado laboral, y teniendo en cuenta el Libro Blanco de la titulación y los recursos humanos y materiales disponibles, se decidió proponer un único título de Grado en Ingeniería Informática, con cinco especialidades profesionales diferenciadas y elevado atractivo en el entorno socioeconómico, regional, nacional y europeo. Las especialidades son:

- Computación y Sistemas Inteligentes
- Ingeniería de Software
- Ingeniería de Computadores
- Sistemas de Información
- Tecnologías de la Información

Los especialistas en Tecnologías de la Información son los encargados de configurar, mantener, actualizar, y garantizar el funcionamiento seguro de las plataformas de cómputo en las empresas. Concretamente, algunos de los temas de estudio de esta última especialidad son la consolidación de servidores a través de técnicas de virtualización o las posibilidades que abre el Cloud Computing [6,7].

Una de las asignaturas obligatorias de la especialidad de Tecnologías de la Información es Servidores Web de Altas Prestaciones.

Esta asignatura tiene como objetivo principal que el alumno aprenda a diseñar, instalar y configurar un servidor de altas prestaciones a partir de las necesidades de un cliente. Los contenidos generales de la asignatura son los siguientes: evaluación de necesidades; requisitos hardware para servidores; criterios de coste, calidad y prestaciones; seguridad en servidores; prestaciones (tolerancia a fallos, alta disponibilidad, etc); ejemplos y aplicaciones.

Con estos contenidos se pretende identificar los problemas más habituales relativos a las aplicaciones, servidores, red y arquitectura, de servidores mal planificados o mal configurados.

En cuanto a los objetivos formativos de la asignatura y los distintos niveles de profundización, capacidades y destrezas que debe adquirir un alumno tras la superación de la asignatura, expresados como resultados esperables de la enseñanza, cabe destacar los siguientes:

- Explicar y exponer los principales conceptos relacionados con la alta disponibilidad, redundancia y tolerancia a fallos.
- Determinar posibles problemas de escalabilidad de una instalación.
- Configurar un balanceador de carga de forma adecuada a las necesidades.
- Realizar las tareas de la administración de un sistema de alta disponibilidad.
- Señalar y describir las tecnologías hardware actuales para la instalación de granjas o agrupaciones de servidores.
- Encontrar y aprovechar las soluciones idóneas para servidores de alta disponibilidad.
- Estimar los costes de diseño, montaje, configuración, mantenimiento y operación del sistema.
- Seleccionar diferentes herramientas de seguridad y describir su uso.
- Establecer la configuración de los servidores y las estrategias para optimizar la seguridad del sistema.
- Establecer políticas de calidad y prestaciones del sistema.
- Seleccionar, instalar y usar las herramientas de análisis y monitorización de las prestaciones del sistema.
- Diseñar y configurar un sistema web de alta disponibilidad.
- Diseñar un plan de copias de seguridad y recuperación para resolver cualquier problema en el sistema.

Como competencias específicas de la asignatura se establecieron:

- TI2. Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.
- TI6. Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.

- TI7. Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.

El resto del trabajo está organizado como sigue: la Sección 2 justifica la necesidad de la asignatura en el plan de estudios. En la Sección 3 se presentan los temarios de teoría y prácticas, seguido de la metodología docente a seguir (Sección 4). Finalmente se detalla la estrategia de evaluación en la Sección 5, para terminar con unas conclusiones en la Sección 6.

2 Justificación de la necesidad de la asignatura en el Grado de Informática

Cuando nos encargan el desarrollo de un sitio web, en muchas ocasiones comenzamos el desarrollo sobre una máquina que hará todas las tareas necesarias (servidor web, almacenamiento, bases de datos, etc). Es la opción más rápida, y si no hay muchos usuarios o el tráfico es bajo, suele funcionar.

Este tipo de sistemas sólo necesita una máquina con el sistema operativo configurado (ya sea, Linux, Windows, OS X o Solaris) una aplicación servidor-web (como Apache, nginx o IIS) y una conexión a internet. Estos sistemas resultan muy baratos inicialmente, por lo que muchas empresas optan por esta configuración. Además, como decimos, es suficiente para sistemas muy pequeños con poco tráfico.

En la misma máquina se instala el software necesario para el servidor web, servidor de mensajería, la base de datos y el almacenamiento de ficheros. A pesar del bajo coste y de que todo vaya bien en la fase de desarrollo y pruebas (por el bajo tráfico), ese servidor puesto en producción puede acabar experimentando diversos problemas.

De hecho, cuando los cambios se vuelven inmanejables, no queda más remedio que reestructurar todo el sistema para configurar una arquitectura basada en granja web (ver Figura 1) que nos permita dar el servicio esperado y sea fácil de mantener.

No merecerá la pena comenzar a parchear el sistema en varios niveles, invirtiendo tiempo, recursos humanos e incluso dinero en adquirir hardware más nuevo y potente: tratar durante meses de adaptar o retocar el sistema para que siga funcionando e intente dar el servicio que debe, no valdrá la pena, ya que saldrá más caro que rediseñar un buen sistema basado en una granja web.

Para ello, y como en casi todo sistema informático, hay que definir correctamente los requisitos del sitio web a desarrollar, invertir el dinero necesario de forma efectiva, y planificar las tareas de configuración y desarrollo de las aplicaciones y servicios de forma adecuada.

En esta asignatura estudiaremos los principales problemas en la construcción y configuración de este tipo de sistemas: problemas de la arquitectura del sistema, con la disponibilidad y con la carga del sistema, principalmente. Asimismo, daremos una visión detallada de los conceptos y herramientas utilizadas para crear o mantener la infraestructura de un sistema de forma adecuada. Estudiaremos las ideas y prácticas para asegurar la escalabilidad del sistema, asegurando el servicio a nuevos usuarios, así como para mantener el sistema activo y dando servicio el máximo tiempo posible.

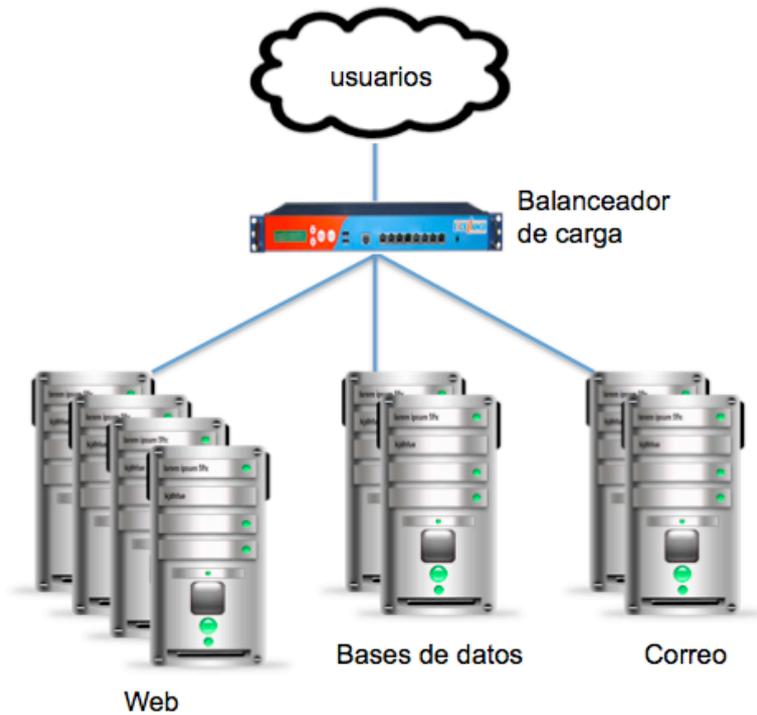


Figura 1. Estructura general de una granja web.

3. Temarios de teoría y prácticas

De acuerdo con los objetivos descritos, se ha planteado el siguiente temario para la asignatura:

- El Tema 1 "Introducción" se dedicará al estudio de los problemas que podemos experimentar al montar un sitio web que tendrá muchos usuarios y un alto tráfico en una máquina sola. De esta forma justificaremos la necesidad de montar granjas web en estos casos.
- El Tema 2 "Alta disponibilidad y escalabilidad en servidores web" presentará los conceptos de alta disponibilidad, escalabilidad y tolerancia a fallos en

general. El objetivo siempre es crear sistemas que den servicio las 24h con el mínimo de errores.

- El Tema 3 "La red de una granja web" presenta diferentes arquitecturas de red para crear una granja web, siempre con los objetivos de rendimiento, disponibilidad y facilidad de mantenimiento [1].
- El Tema 4 "Balanceo de carga" detalla diversas opciones de balanceo de carga, tanto software como hardware [2,5].
- El Tema 5 "Medición de prestaciones" presenta algunas consideraciones sobre cómo evaluar las prestaciones de un sistema web de altas prestaciones.
- El Tema 6 "Técnicas de seguridad" presenta los diferentes problemas de seguridad a tener en cuenta para asegurar los servidores. Se comentan las prácticas comunes para resolver problemas de seguridad, el uso de cortafuegos, protección de servidores y políticas de seguridad.
- En el Tema 7 "Requisitos hardware de servidores web" se estudian los principales requisitos a nivel de hardware para la construcción y configuración de estos sistemas
- De forma similar, el Tema 8 "Requisitos software de servidores web" estudia los requisitos a nivel de software.
- En el Tema 9 "Bases de datos de alta disponibilidad" se detallan las cuestiones principales relativas a la configuración de bases de datos en sistemas de alta disponibilidad [4].
- Por último, el Tema 10 "Aplicaciones y ejemplos" presenta diferentes sistemas usados en empresas u organizaciones en las que se hace uso de este tipo de sistemas.

Igualmente, se han propuesto, como complemento docente, los siguientes seminarios:

- Seminario práctico 1: Componentes y configuración hardware de un cluster web
- Seminario práctico 2: Alternativas de sistemas de balanceo de carga
- Seminario práctico 3: Alternativas para clustering en servidores web
- Seminario práctico 4: Configuración de sistemas y técnicas de seguridad en servidores web
- Seminario práctico 5: Alternativas en bases de datos de altas prestaciones

Finalmente, la parte práctica de la asignatura está compuesta por las siguientes prácticas:

- Práctica 1: Presentación de las prácticas y preparación de las herramientas
- Práctica 2: Replicación de contenidos entre servidores
- Práctica 3: Balanceo de carga y clustering en servidores web
- Práctica 4: Evaluación de prestaciones
- Práctica 5: Uso de bases de datos en sistemas de alta disponibilidad

En las siguientes secciones se presenta de forma más detallada la metodología prevista para el desarrollo de la asignatura (teoría y prácticas), así como la estrategia de evaluación.

4. Metodología propuesta

El desarrollo de la asignatura se basará en los siguientes tipos de actividades de formación:

- Clases teóricas para la presentación en el aula de los conceptos propios de la materia, haciendo uso de metodología expositiva con lecciones magistrales participativas y medios audiovisuales.
- Clases prácticas, a través de las cuales se pretende mostrar al alumnado cómo debe actuar a partir de la aplicación de los conocimientos adquiridos.
- Seminarios para tratar en profundidad temáticas relacionadas con la materia. Se trata de actividades basadas en la indagación, el debate, la reflexión y el intercambio.
- Estudio y trabajo en grupo: Se trata de trabajos propuestos por el profesor, a través de los cuales y de forma grupal se profundiza en aspectos concretos de la materia, posibilitando a los estudiantes avanzar en la adquisición de determinados conocimientos y procedimientos de la materia.

Las actividades formativas propuestas más arriba se desarrollarán desde una metodología participativa y aplicada que se centra en el trabajo del estudiante (presencial y no presencial/individual y grupal). Las clases teóricas, los seminarios, las clases prácticas, las tutorías, el estudio y trabajo autónomo y el grupal son las maneras de organizar los procesos de enseñanza y aprendizaje.

4.1 Herramientas utilizadas en la asignatura

Un aspecto importante son las herramientas utilizadas en el desarrollo de la asignatura. En este sentido, un aspecto fomentado es la creación de blogs de alumnos, en los que se describan las inquietudes, intereses, enlaces y trabajos de los mismos en relación a la asignatura. Para ello, se insta a que éstos hagan la entrega de sus prácticas a modo de página web, consiguiendo igualmente un formato legible y bien presentado en cualquier sistema. Esta medida (probada anteriormente en otras

asignaturas) suele tener un gran acogimiento entre los alumnos, manteniendo muchos de los cuales dicho blog en adelante.

La otra herramienta utilizada es SWAD: Sistema Web de Apoyo a la Docencia(<http://swad.ugr.es>) [3], una plataforma libre de teleformación desarrollada y utilizada en la Universidad de Granada en los últimos 12 cursos académicos.

SWAD integra diversas funciones de apoyo al aprendizaje, a la docencia y a la gestión de los datos de los estudiantes. Entre ellas el acceso a información sobre las asignaturas (guía docente, horarios, bibliografía,...), la descarga de documentos (transparencias, relaciones de problemas), las listas y fichas de alumnos y profesores, los foros de discusión, la asignación de actividades, la autoevaluación mediante exámenes interactivos o la consulta individual de calificaciones. En dicha plataforma se hace el seguimiento de los alumnos, la publicación de los temarios y guiones de prácticas, la entrega de ejercicios, trabajos y prácticas, y la publicación (con consulta individual) de las calificaciones.

5. Evaluación

Con objeto de evaluar la adquisición de los contenidos y competencias, se utilizará un sistema de evaluación diversificado, seleccionando las técnicas de evaluación más adecuadas para cada materia o asignatura en cada momento, que permita poner de manifiesto los diferentes conocimientos y capacidades adquiridos por el alumnado al cursar la materia.

Se utilizarán las siguientes técnicas de evaluación:

- Para la parte teórica se realizará un examen final tipo test así como entregas de ejercicios y resultados de las actividades propuestas.
- Para la parte práctica se realizarán prácticas de laboratorio y desarrollo de proyectos (individuales o en grupo), y se valorarán las entregas de los informes/memorias realizados por los alumnos.
- En cuanto a los seminarios, estos se evaluarán teniendo en cuenta la asistencia y participación.
- En cuanto a la parte de trabajo en grupo, se evaluará teniendo en cuenta la presentación oral así como la memoria del trabajo desarrollado.

El resultado de la evaluación será una calificación numérica obtenida mediante la suma ponderada de las calificaciones correspondientes a la parte teórica, la parte práctica y una parte relacionada con el trabajo autónomo de los alumnos, los seminarios impartidos y el aprendizaje basado en proyectos.

6. Conclusiones

El objetivo de este trabajo es la presentación de la asignatura Servidores Web de Altas Prestaciones, de la especialidad Tecnologías de la Información del nuevo plan de estudios del Grado en Ingeniería Informática de la Universidad de Granada.

La metodología docente está pensada para impartir la asignatura de forma práctica siempre que sea posible. En cuanto a las herramientas docentes que se usarán, se proponen SWAD, una plataforma web para facilitar las tareas de docencia, y diversos blogs de alumnos que versen sobre la asignatura y sus contenidos. Los años de experiencia usando estas herramientas en otras asignaturas y la metodología propuesta han demostrado que el método funciona, y que los alumnos adquieren y afianzan conocimientos.

Por último, por su temática, cabe destacar la necesidad de mantener actualizados los temarios de teoría y prácticas con las últimas tecnologías, herramientas y metodologías de estudio en la construcción y configuración de servidores web de altas prestaciones.

Agradecimientos

Los autores agradecen a Mario Barchéin sus aportaciones y ayuda en la preparación de la asignatura.

Referencias

- [1] B. Bloom. *Deploying and Managing Microsoft .NET Web Farms*. ISBN 0672320576. Sams White Bks. Indianapolis, IN, USA. 2001
- [2] T. Bourke. *Server Load Balancing*. 1 Edición. ISBN 0-596-00050-2. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, USA. 2001
- [3] A. Cañas, D. J. Calandria, E. M. Ortigosa, E. Ros, A. F. Díaz. *SWAD: Web System for Education Support*. In B. Fernández-Manjón, J. M. Sánchez Pérez, J. A. Gómez-Pulido, M. A. Vega Rodríguez, J. Bravo-Rodríguez (Eds.): *Computers And Education: E-learning - from Theory to Practice*, 241 pages, Chapter 12, pp. 133-142, ISBN 978-1-4020-4913-2, Springer, 2007.
- [4] A. Davies. *High Availability MySQL Cookbook*. ISBN 978-1-847199-94-2. Packt Publishing. 2010
- [5] C. Kopparapu. *Load Balancing Servers, Firewalls, and Caches*. ISBN 0-471-41550-2. Wiley Computer Publishing John Wiley & Sons, Inc. USA. 2001.

- [6] M. Miller. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. ISBN-10: 0789738031. ISBN-13: 978-0789738035. 1ª Edición. QUE. 2012
- [7] B. Furht and A. Escalante. Handbook of Cloud Computing. Google eBook. ISBN: 978-1-4419-6523-3. Springer. 2010

Instrucciones para Autores

Enseñanza y Aprendizaje de Ingeniería de Computadores (Teaching and Learning Computer Engineering) es una revista de Experiencias Docentes en Ingeniería de Computadores que edita el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada, se publica anualmente, y se difunde tanto en papel como electrónicamente, a través del repositorio institucional de la Universidad de Granada (<http://digibug.ugr.es/>).

Los artículos remitidos para su evaluación pueden estar escritos en castellano o inglés, incluyendo un resumen y palabras clave en inglés en caso de que estén escritos en castellano, y deben seguir el formato descrito en la dirección web:

http://atc.ugr.es/pages/actividades_extension/

El correspondiente fichero .pdf debe enviarse a la dirección de correo electrónico jortega@ugr.es o mdamas@ugr.es

Los artículos deben abordar, tanto contenidos relacionados con la docencia universitaria en general, como con la docencia de asignaturas específicas impartidas por las áreas de conocimiento involucradas en estudios relacionados con la Ingeniería de Computadores, y también pueden aspectos relativos a las competencias profesionales y la incidencia de estos estudios en el tejido socio-económico de nuestro entorno.

En particular, se anima a antiguos alumnos de los estudios de Informática y a estudiantes de grado y posgrado a que envíen colaboraciones relacionadas con sus experiencias al cursar asignaturas relacionadas con la Ingeniería de Computadores, sugerencias, propuestas de mejora, etc.

Teaching and Learning Computer Engineering

**Journal of Educational
Experiences on Computer
Engineering**

May 2013, Number 3

