

# Integración de TICS en la Docencia Teórica de Una Asignatura de Programación Concurrente

Antonio Tomeu  
University of Cádiz

Computer Science Department  
C/Chile s/n. 11002, Cádiz (Spain)  
antonio.tomeu@uca.es

Alberto Salguero  
University of Cádiz

Computer Science Department  
C/Chile s/n. 11002, Cádiz (Spain)  
alberto.salguero@uca.es

Manuel Capel  
University of Granada

C/Periodista Daniel Saucedo Aranda  
18015, Granada (Spain)  
manuelcapel@ugr.es

## ABSTRACT

Teaching computer programming usually implies to show source code listings, in a given programming language, to supports the explanations of the teacher. Traditionally, these source codes have been presented to the students using the classroom board, slides or projectors, being accompanied by the pertinent oral clarifications of the teacher. The students do not test the code or make any amendment that reinforces the concept being illustrated. The proposal presented in this project is to change the traditional expositive model usually used in programming courses, by another in which the students become more participative. In this new model the students use a laptop during the lectures to immediately test the source code listings available in the virtual campus, with the help of the teacher.

## RESUMEN

La enseñanza de la programación de ordenadores requiere, durante la exposición teórica, la presentación al alumno de código en un lenguaje de programación dado, que dé soporte al cumplimiento de un especificación concreta. Tradicionalmente, dichos códigos han sido presentados al alumno en pizarra, diapositiva o proyector, realizando el profesor una descripción oral de su funcionamiento. Pero el alumno ni prueba el código ni hace modificaciones al mismo que afiancen el concepto que se pretende ilustrar con él. La propuesta que se presenta en este proyecto es el cambio del modelo expositivo tradicional en las asignaturas de programación, por otro en que el alumno disponga durante la clase de teoría de una ordenador portátil para que, utilizando el soporte proporcionado por el profesorado mediante el Campus Virtual, puede poner inmediatamente en práctica los códigos con los cuales se le presentan los conceptos.

## Palabras clave

Innovación Docente, Mejora, Campus Virtual, Integración de TIC, Ordenadores en Clase, Modelo de Docencia Teórica, Programación Concurrente, Código, Mejora de Rendimiento, Clases de Teoría, Alumnos.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference'10, Month 1-2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

## 1. INTRODUCCION

En la actualidad, cualquier computadora que adquirimos viene equipada con un procesador de múltiples núcleos que permite al programador efectuar concurrencia real o si se quiere, paralelismo. Los actuales currícula de Grado en Ingeniería Informática han sido sensibles a esta nueva realidad, y han considerado la importancia de capacitar al futuro graduado en el uso de la técnicas de Programación Concurrente que le permitan explotar el paralelismo que los múltiples núcleos ofrecen.

Estos contenidos, que tradicionalmente formaban parte de asignaturas optativas, son ahora de obligado conocimiento por los estudiantes. Sin embargo este cambio, a todas luces positivo y necesario, no ha venido, en general, acompañado de una nueva forma de enseñar ni la programación ni la concurrencia. A pesar de los nuevos modelos de docencia que el EEES preconiza, enseñar programación sigue requiriendo exponer y analizar código, y esto no es susceptible de modificarse. Lo que sí puede modificarse es cómo hacer que el alumno analice ese código, y juegue un papel protagonista -en lugar del tradicional papel pasivo- en el mismo. El proyecto de innovación docente que describe esta memoria expone cómo lograr tal cambio, y los resultados obtenidos de la aplicación de un nuevo modelo de docencia en un caso real.

### 1.1 Contexto

Los siguientes *ítems* describen el marco contextual en el que ha tenido lugar el desarrollo de la experiencia:

- La experiencia ha tenido lugar durante el actual curso 2012/2013 durante el que se ha desarrollado la docencia de la asignatura para un total de 149 alumnos matriculados, estructurado en dos grupos teóricos de 75 alumnos cada uno. Se imparte durante el primer cuatrimestre del curso con un total de 3 créditos teóricos y 3 créditos prácticos.
- El proyecto ha sido desarrollado por los dos docentes asignados a la docencia de la asignatura, pertenecientes al mismo departamento y a diferentes áreas de conocimiento: Ciencias de la Computación e Inteligencia Artificial, y Lenguajes y Sistemas Informáticos.
- Con objeto de atender a la especial infraestructura necesaria para efectuar al desarrollado pretendido de la docencia teórica, se solicitó a la dirección de la Escuela Superior de Ingeniería el uso de un aula dotada de ordenadores portátiles para uso en depósito

durante la clase de los alumnos, mesas con alimentación disponible para aquellos alumnos que harían uso de su propio portátil, y acceso a la red inalámbrica de la Universidad. Se tuvo también disponible un cañón proyector y una pizarra, junto con la habitual mesa multimedia para uso docente.

- La docencia teórica se desarrolló durante el calendario previsto por el Centro durante el cuatrimestre, en sesiones teóricas de dos horas semanales, de acuerdo a lo establecido por la Subdirección de Ordenación Académica del Centro.

## 1.2 Objetivos

Los objetivos que perseguimos con la experiencia que se describe son los siguientes:

- Una mejora significativa de la exposición de la docencia teórica de la programación concurrente en particular, y de la programación en general.
- La integración de las TIC en el ámbito de la docencia teórica, más allá del uso de soportes como el cañón proyector o la pizarra electrónica que, por sí mismos, no cambian el carácter pasivo del alumno durante la fase expositiva, mutando ese carácter por uno netamente activo.
- Un incremento de la interactividad del trabajo del alumno en el aula durante una clase de teoría.

## 1.3 Cronograma

La experiencia se ha desarrollado de conformidad con el cronograma que se muestra en la Figura 1.

ACTUACIÓN	jun	jul	ago	sep	oct	nov	dic	ene	feb	mar	abr	may	jun	jul
Elaboración de Código	■	■	■											
Preparación del Campus	■	■	■	■										
Docencia					■	■	■	■	■	■				
Elaboración de Encuesta							■	■						
Recogida de Datos									■	■				
Análisis de Resultados											■	■	■	■
Elaboración de Memoria													■	■
Entrega de Memoria														■

Figura 1. Cronograma del Proyecto.

La práctica totalidad de los hitos temporales recogidos en la misma se explican por sí solos en el contexto del documento, no requiriéndose más análisis sobre la misma.

## 2. QUÉ SE INNOVA

La docencia de la programación en general, y de la Programación Concurrente en particular, tradicionalmente se venía realizado exponiendo un código de programa expuesto en pizarra, diapositiva o cañón proyector (en la Figura 2 se muestra un ejemplo de uso de una diapositiva de las utilizadas habitualmente por nosotros con cañón proyector), sobre el que el docente explicaba los conceptos y el funcionamiento para inmediatamente pasar al análisis del siguiente código sobre la diapositiva siguiente. El alumno quedaba con una visión incompleta del contenido semántico del mismo, puesto que no probaba la depuración y ejecución de manera inmediata del código propuesto.

```

CONTROL DE Threads: Ejemplo de Cesión de Prioridad Voluntaria (yield)
public class replaniYield
extends Thread
{
    private boolean hY;//indicara si el hilo cede prioridad o no.
    private int    v;

    public replaniYield(boolean haceryield, int vueltas)
    {hY = haceryield; v = vueltas;}

    public void run()
    {
        for(int i=0; i<v; i++)
        if(i%20&&hY==true){this.yield();};//indica cesion de prioridad.
        else System.out.println("Hilo "+this.getName()+" en iteracion "+i);
    }
    public static void main(String[] args)
    {
        replaniYield h0 = new replaniYield(false, 50);
        replaniYield h1 = new replaniYield(false, 50);
        replaniYield h2 = new replaniYield(true , 50); //cedera prioridad y
        h0.setName("1-NoYield"); //sera o no considerada
        h1.setName("2-NoYield"); h2.setName("3-SiYield");
        h0.start(); h1.start(); h2.start();
    }
}
    
```

© Antonio Tomeu Creación y Control de Thread en Java 27

Figura 2. Ejemplo de Diapositiva.

El carácter pasivo del alumno durante la exposición quedaba limitado, en el mejor de los casos, porque solicitaba alguna aclaración concreta.



Figura 3. Nuevo Modelo de Docencia Teórica.

La propuesta que presentamos pasa por una reformulación radical del papel del alumno durante una sesión de docencia teórica de programación. Para ello, el alumno debe disponer durante la sesión de un ordenador portátil o de sobremesa con acceso al Campus Virtual (ver Figura 3). En el mismo, los docentes de la asignatura hemos diseñado el material y el soporte de código de programa necesario para que el alumno pueda disponer del código que el profesor está explicando sobre una diapositiva de forma inmediata a los efectos de descarga, depuración -en su caso-, ejecución y prueba.

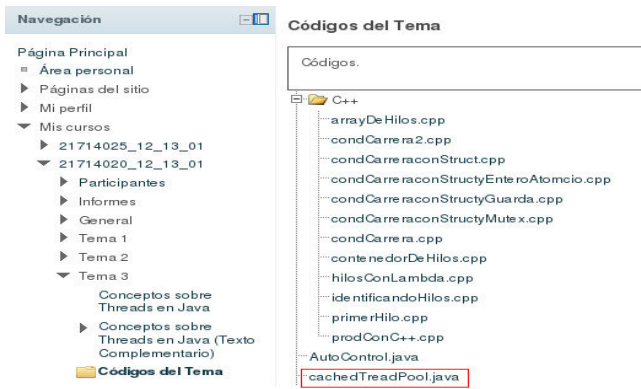
Para ello, cada bloque del Campus Virtual correspondiente a un tema del curso, incluye una carpeta de códigos, tal y como es muestra en la Figura 4. Una vez descargado el código, el alumno desarrolla el nuevo ciclo de trabajo que proponemos para la docencia teórica, y cuyo flujo de tareas mostramos en la Figura 6. Como se aprecia en ella, el alumno deja de ser el receptor pasivo de la información que sobre un código de programa le transmite su profesor, para asumir el protagonismo absoluto en el análisis empírico del mismo mediante su descarga desde el Campus Virtual, depuración, ejecución y análisis de las salidas.

**TEMA 3: CREACIÓN Y CONTROL DE THREADS EN JAVA**



**Figura 4. Un tema en el campus.**

Si tras esta fase el grupo ha asimilado bien el concepto se proponen ejercicios de refuerzo que son realizados de forma individual. Los códigos resultantes de esos ejercicios son enviados a un foro del Campus Virtual específicamente destinado para ello, tras lo cual se realiza una puesta en común de los posibles enfoques de la solución pedida.



**Figura 5. La Carpeta de Códigos.**

**2.1 Metodología de Desarrollo**

Durante la elaboración de contenidos necesarios para la implementación del proyecto de innovación, la pauta metodológica de trabajo seguida por los profesores de la asignatura ha seguido las fases siguientes:

- Para cada tema del curso, los profesores han elaborado un conjunto de códigos de programa que ilustran los conceptos teóricos que se pretenden desarrollar.
- Los códigos, previa revisión cruzada entre los profesores, se han puesto a disposición de los alumnos con carácter semanal, mediante la carpeta específica del Campus Virtual que ya hemos comentado (ver Figuras 4 y 5). Se han dispuesto dos versiones de código, una compuesta por ficheros de texto plano, y otra con el conjunto de ficheros comprimido para más fácil descarga.
- Durante la exposición teórica, una vez que el profesor ha presentado el código de programa e ilustrado su funcionalidad, se ha invitado al alumno seguir la pauta de trabajo descrita en el diagrama de flujo de la Figura 6.

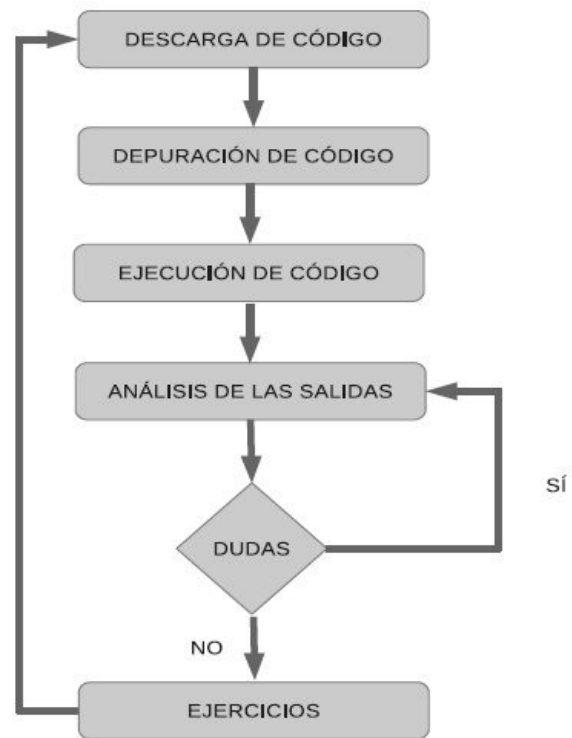
**3. RESULTADOS**

Para el análisis de los resultados de la experiencia se ha escogido un doble escenario prospectivo, desarrollando acciones de análisis respectivamente previa y posteriormente a la evaluación de la asignatura:

**3.1 Preevaluación**

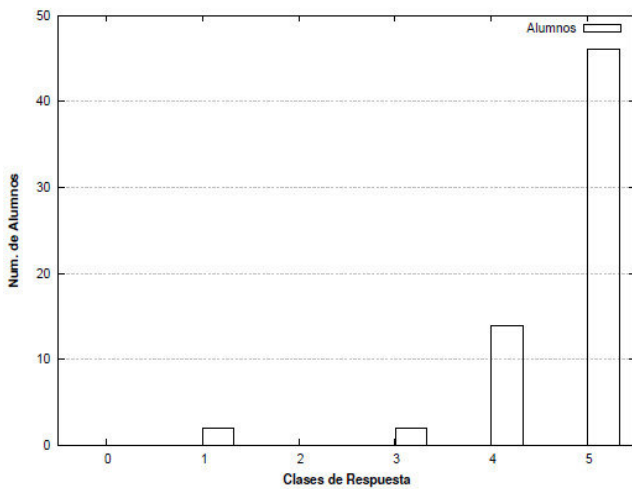
Con objeto de conocer las impresiones subjetivas que el nuevo modelo de docencia teórica de la asignatura estaba dejando en los alumnos matriculados mientras se les aplicaba, se diseñó una pequeña encuesta que tenía por objetivo, con tres meses consolidados de docencia, medir cuantitativamente las siguientes variables:

- Mejora en la comprensión de los conceptos expuestos durante las sesiones de docencia teórica.
- Adecuación del número de ejercicios propuestos en clase para afianzar los conceptos expuestos.
- Adecuación del tiempo dedicado a la resolución de los ejercicios propuestos.
- Grado de conformidad con el nuevo modelo de docencia teórica.



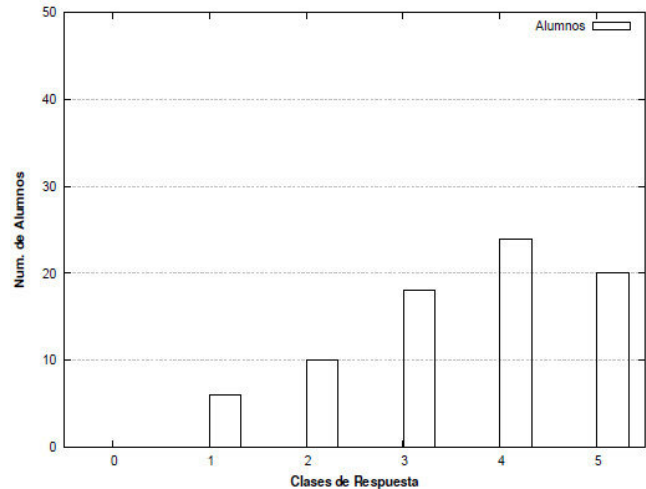
**Figura 6. Ciclo de Trabajo con el Nuevo Modelo.**

Cada uno de los ítems anteriores podía valorarse con una puntuación de entre uno (totalmente en desacuerdo) y 5 (totalmente de acuerdo) con un sexto ítem adicional para el caso de que el alumno no quisiera contestar. Los siguientes histogramas muestran los resultados obtenidos.



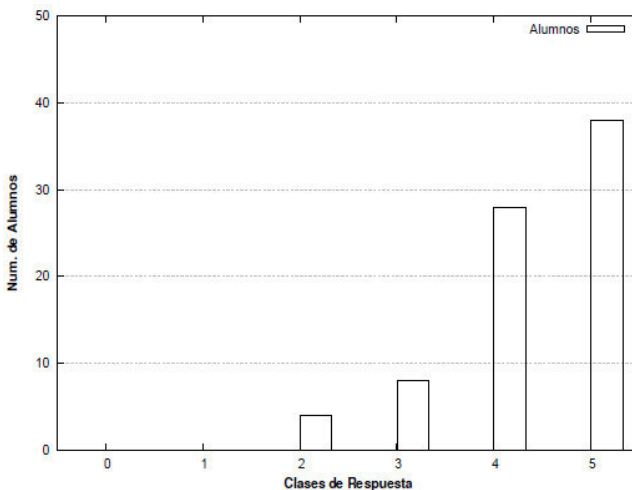
**Figura 7. Mejora en la Comprensión.**

El histograma<sup>1</sup> de la Figura 7 recoge la opinión de los alumnos sobre el grado de mejora de la comprensión de los conceptos de programación expuestos durante las sesiones de docencia teórica utilizando el nuevo modelo. Vemos como la práctica totalidad de los alumnos están totalmente de acuerdo o muy de acuerdo en el beneficio que el modelo les aporta frente al esquema de docencia teórica pasivo clásico.



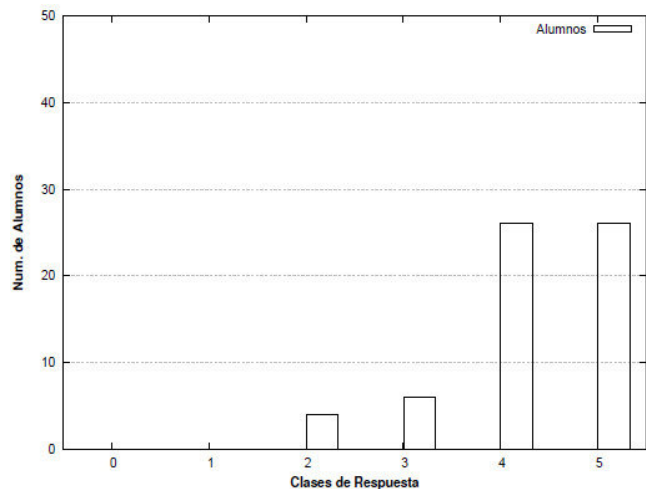
**Figura 9. Adecuación Tiempos de Resolución.**

El histograma de la Figura 9 recoge la opinión de los alumnos acerca del tiempo que se les permite utilizar para el desarrollo de la solución a los ejercicios de afianzamiento de conceptos propuestos, donde apreciamos que aproximadamente una tercera parte de los alumnos considera que ese tiempo es inadecuado. De nuevo no sabemos si por exceso o por defecto, y será necesario realizar una investigación posterior sobre este aspecto para concretar posibles acciones de mejora.



**Figura 8. Adecuación Número De Ejercicios.**

El histograma de la Figura 8 recoge la opinión de los alumnos acerca de la fase de ejercicios del diagrama de la Figura 6. En concreto, se desea conocer si el número de ejercicios propuestos es el adecuado para afianzar un concepto tras haberlo presentado. De nuevo la gran mayoría de los alumnos considera que es así, aunque en esta ocasión los datos están algo más dispersos. Un pequeño grupo de alumnos considera inadecuada ese número, aunque no podemos discriminar si por exceso o por defecto.



**Figura 10. Grado Acuerdo con el Modelo.**

El histograma de la Figura 10 recoge la opinión de los alumnos acerca del grado de acuerdo con el nuevo modelo de docencia teórica experimentado, y de nuevo la gran mayoría de alumnos se decantan entre estar “de acuerdo” o “muy de acuerdo” con él. Comprobamos así que en los dos ítems de mayor interés para validar la viabilidad del nuevo modelo de docencia teórica (mejora de la comprensión y grado de acuerdo con el nuevo modelo), los resultados obtenidos durante la experiencia han sido en general muy buenos.

### 3.2 Postevaluación

En este caso se trata de una medida objetiva del grado de bondad del nuevo modelo de docencia mediante un análisis de rendimiento comparado del actual curso académico 2012/2013 con el curso anterior. Para efectuar el análisis se comparó el

<sup>1</sup> La leyenda de los histogramas representa la ausencia de respuesta mediante 0, clasificando las respuestas en la escala ya indicada: de 1 para “totalmente en desacuerdo” hasta 5 para “totalmente de acuerdo”.

número de alumnos clasificados en las diferentes categorías de calificación entre ambos cursos, obteniéndose el resultado que se muestra en el histograma de la Figura 11<sup>2</sup>.

La figura describe en % sobre matriculados el número de alumnos no presentados (valor 0), y en % sobre presentados los alumnos suspensos, aprobados, notables y sobresalientes (resto de valores). Se aprecia una mejora significativa<sup>3</sup> entre los resultados obtenidos en el curso 2011/2012 y el actual 2012/2013. En concreto, hay una disminución significativa entre el número de alumnos que no se presentaban a examen o que, presentándose, lo suspendían. Y hay también una evolución positiva destacable en el número total de alumnos que superan la asignatura, y entre las diferentes subcategorías de calificación dentro del grupo de los que la superan. Estos datos que comparan ambos cursos tras la evaluación vienen a corroborar los que ya habíamos presentado con carácter subjetivo en la anterior sección, y las impresiones que los alumnos mostraron allí: a la sensación de mejora de la comprensión y del grado de acuerdo con el modelo, ahora se añade una evolución positiva en el rendimiento académico de esos mismos alumnos, que permite por tanto validar el nuevo modelo de docencia teórica de la programación.

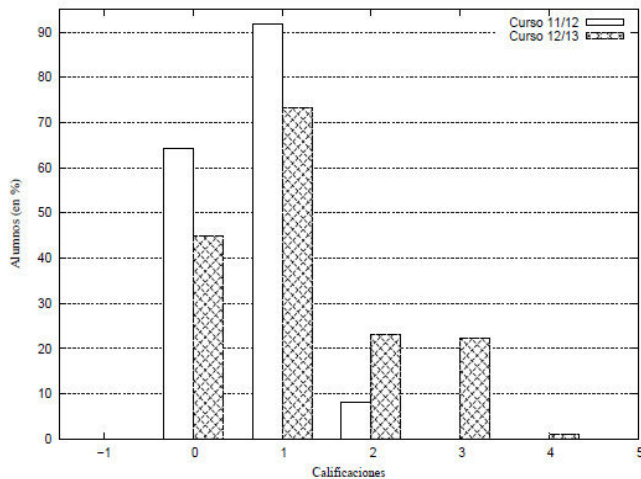


Figura 11. Rendimiento Comparado Intercurso.

#### 4. CONCLUSIONES

Tras el análisis de los resultados de la experiencia, en función a los resultados obtenidos durante las fases de preevaluación y postevaluación, hemos llegado a la conclusión de mantener el modelo de docencia teórica experimentado en los próximos cursos académicos, puesto que:

- Se han alcanzado los objetivos que nos habíamos propuesto como meta.

<sup>2</sup> La leyenda de este histograma es: 0-No Presentado, 1-Suspense, 2-Aprobado, 3-Notable y 4-Sobresaliente.

<sup>3</sup> Las cifras que se ofrecen corresponden a la convocatoria de Febrero de la asignatura en los cursos 11/12 y 12/13. En el momento de la redacción de este documento no se dispone de las cifras de la asignatura para las convocatorias de Junio y Septiembre del curso 12/13. Es de prever una evolución positiva similar en las distintas categorías de calificación.

- Se ha mejorado la comprensión de los conceptos de programación que se presentan al alumno.
- Se ha migrado de una realidad donde el alumno es un ente pasivo durante una clase de teoría a otra en la cual se ve obligado a tener una actitud proactiva.
- De manera indirecta, se ha logrado que el alumno haga un seguimiento continuo de los contenidos de la materia, puesto que ya no se trata de venir a clase a escuchar al profesor y -eventualmente- tomar algunas notas. Ahora necesita un dominio actualizado de los contenidos, en orden a un eficaz aprovechamiento del nuevo modelo de docencia teórica.
- Igualmente de manera indirecta, el alumno llega más preparado para afrontar la docencia práctica de la asignatura; ellos nos ha permitido aumentar la carga de trabajo a desarrollar durante las sesiones prácticas, así como la complejidad de las mismas.
- Se ha conseguido que el alumno se implique de forma más personal y rigurosa en el quehacer diario de la asignatura.
- Se ha logrado integrar de forma radical el uso de las TIC en la docencia teórica de la programación, de forma eficaz y no traumática para el alumno.
- La tasa de éxito de los alumnos de la asignatura ha mejorado notablemente frente al anterior curso, donde se siguió el modelo tradicional.
- Y finalmente, y hay que tenerlo muy en cuenta, los alumnos nos dicen que “nos gustan vuestras clases de teoría así”.

#### 5. AGRADECIMIENTOS

Agradecemos a la Subdirección de Ordenación Académica de la Escuela Superior de Ingeniería, las facilidades proporcionadas para situar la docencia teórica que era objeto del proyecto de innovación en el aula adecuada para ello.

#### 6. REFERENCIAS

- [1] Area Moreira, M. *Enseñar y aprender con TIC: más allá de las viejas pedagogías*. Aprender a educar con tecnología, nº2, diciembre 2012, pgs. 4-7.
- [2] Area Moreira, M. *Una breve historia de las políticas de incorporación de las tecnologías digitales al sistema escolar en España*. Quaderns digitals: Revista de Nuevas Tecnologías y Sociedad, ISSN-e 1575-9393, N°. 51, 2008.
- [3] Bloom, B.S., et al. *Taxonomy of Educational Objectives: Handbook I, Cognitive Domain*. New York: David McKay, 1956.
- [4] Schechter, E.I. *Internet Resources for Higher Education Outcomes Assessment*. JFECS, 8 (2), 105-107.
- [5] Vilela, A. *Moodle 2 Para Profesores*. Ed Rom, 2009.