

# Diseño y evaluación de la asignatura Programación I de acuerdo a las directrices del Espacio Europeo de Educación Superior

Jesús Ibáñez Martínez, Toni Navarrete Terrasa

Dpto. de Tecnología  
Universitat Pompeu Fabra  
Passeig de Circumval·lació, 8 – 08003 Barcelona  
{jesus.ibanez , toni.navarrete}@upf.edu

## Resumen

Este artículo describe el nuevo diseño de la asignatura Programación I en sintonía con las directrices del nuevo Espacio Europeo de Educación Superior. Los objetivos que han guiado el diseño son: reducción del número de sesiones de teoría, reducción del salto entre teoría y práctica, incremento del trabajo efectivo de los alumnos, e incremento de la interacción entre profesores y alumnos. El artículo también detalla el primer estudio llevado a cabo con el fin de evaluar la experiencia. Los resultados de este estudio avalan el éxito del nuevo diseño.

## 1. Contexto y objetivos de la experiencia

La asignatura de Programación I forma parte de las titulaciones de Ingeniería Informática e Ingeniería Técnica en Informática de Sistemas de la Universitat Pompeu Fabra. En ambos casos se cursa durante el primer trimestre del primer curso. Hay que mencionar que en la UPF todas las asignaturas se estructuran por trimestres (y no cuatrimestres) de 10 semanas lectivas y entre 2 y 3 semanas de exámenes. La asignatura se corresponde con 5 créditos, divididos en 3 de teoría y 2 de prácticas. Actualmente la cursan aproximadamente 130 alumnos (incluyendo convalidaciones), que se dividen en dos grupos de teoría y 5 de prácticas, sin distinción entre los estudiantes de las dos titulaciones.

El objetivo principal de esta asignatura es el que el alumno adquiera los fundamentos de la algorítmica y estructuras de datos más simples y aprenda a aplicarlos para resolver problemas. También ha de ser capaz de implementar programas en el lenguaje C y se ha de familiarizar con un entorno de desarrollo. En concreto, los contenidos que se ofrecen en la asignatura son:

- Conceptos generales: compilación e interpretación, programa y algoritmo
- Tipos de datos básicos
- Expresiones, sentencias y estructuras de control
- Descomposición funcional y diseño descendente
- Tipos de datos compuestos estáticos (matrices, cadenas y estructuras)
- Declaración de tipos propios
- Punteros, gestión dinámica de memoria y paso de parámetros por referencia
- Ficheros de texto

Programación I, para la que no se presupone ningún conocimiento previo de algorítmica y programación, es la primera de una serie de asignaturas del bloque de Programación y Estructuras de Datos en el primer curso. Así, esta asignatura es la base para Programación II y Estructuras de Datos y de la Información I, ambas del segundo trimestre, y Programación III en el tercero. En concreto, en Programación II se profundiza en la algorítmica (recursividad, técnicas de diseño y análisis de algoritmos); Estructuras de Datos y de la Información I se centra en los TAD más frecuentes como listas, pilas, colas y ciertos tipos de árboles; y en Programación III se presentan las bases de la programación orientada a objetos y del lenguaje Java. Además, como es habitual en una carrera de Informática, otras asignaturas tienen un alto nivel de interacción con Programación I. Por ejemplo, en el primer curso, en Álgebra y Matemática Discreta II (segundo trimestre) se han de implementar en C unos problemas sobre grafos, o en Análisis Numérico (tercer trimestre) se implementan también en C rutinas para cálculo matricial.

Hasta el curso anterior la asignatura se estructuraba en 30 horas de clase de teoría donde se iban explicando los conceptos y técnicas, y donde tanto alumnos como profesores solucionaban en la pizarra ejercicios que se habían planteado en la sesión anterior. Esta parte de la asignatura se evaluaba mediante un examen al final del trimestre. Por otro lado, los alumnos tenían 10 sesiones de prácticas de 2 horas cada una, donde tenían que resolver problemas más complejos que los expuestos en clase, y que normalmente debían finalizar fuera del aula. En particular, dos prácticas, una a mitad y otra al final del trimestre eran evaluables y suponían el 50% de la nota final de la asignatura. Conviene resaltar que la asignatura es ciertamente densa, ya que se introducen en 10 semanas un número considerable de conceptos y técnicas (a los que en otras facultades se dedican normalmente entre uno y dos cuatrimestres); y también que la estructura es completamente incremental en el sentido de que es imprescindible haber asimilado todos los conceptos y técnicas anteriores para comprender los nuevos.

Teniendo en cuenta estos condicionantes, el principal problema de esta metodología era la prácticamente nula participación de los alumnos en la parte de teoría. Los alumnos generalmente no realizaban los ejercicios planteados, aún cuando el presentarlos en la pizarra se valoraba positivamente para la nota final. De esta forma, el aprovechamiento de las sesiones de teoría era muy bajo. Y por otro lado, el salto entre los ejercicios de teoría que no habían hecho y los de prácticas se convertía en mucho mayor del deseado, lo que conllevaba un importante número de abandonos en las prácticas, especialmente en la final. El hecho de ser una asignatura de primer trimestre de primer curso, donde los alumnos provenientes de bachillerato aún no han adquirido un hábito de trabajo diario, agrava el problema.

Así, el porcentaje de aprobados en diciembre del curso 2004/05 fue de un 33,8% respecto de los presentados, y de alrededor del 20% respecto del global. Estos números, que de hecho iban bajando cada curso, nos llevaron a plantearnos un cambio en la metodología, que permitiera, con el mismo nivel de exigencia, incrementar el porcentaje de alumnos aprobados. Para

conseguir este fin, los objetivos concretos que nos planteamos son: reducir el número de sesiones de teoría “magistrales” en los que la participación de los alumnos es mínima, reducir el salto entre teoría y práctica, incrementar el trabajo efectivo de los alumnos, e incrementar la interacción entre profesores y alumnos.

Como puede observarse, estos objetivos están en la línea de lo que plantean las directrices del nuevo Espacio Europeo de Educación Superior (EEES). Así pues, en esta experiencia nos planteamos ya adaptar completamente la asignatura a las directrices EEES, diseñando un nuevo plan docente basado en competencias, que detalla las actividades de aprendizaje y la dedicación temporal necesaria. Nuestra experiencia recibió una ayuda del Programa para la Calidad Educativa (PQE) de la UPF, en la modalidad de “Experimentación Bolonia”. Es conveniente aclarar aquí que un requisito importante para llevar a cabo la experiencia es que no supusiera un incremento de dotación del profesorado.

## 2. Diseño de la experiencia

Comenzamos revisando el estado del arte en docencia de programación. Para ello nos resultó especialmente útil el trabajo de Costelloe [1]. Buscamos nuevos enfoques y herramientas que, sin requerir cambios en los contenidos de la asignatura, pudiesen contribuir a aumentar el interés de los estudiantes. Consideramos seriamente el uso de herramientas de visualización de programas y robots. Sin embargo, no encontramos ninguna herramienta interesante de visualización de programas en C (aunque sí las encontramos para otros lenguajes como Java, Pascal, etc). La opción de utilizar robots la abandonamos al parecernos arriesgado el diseño de prácticas de programación de robots que cubran adecuadamente los contenidos de la asignatura. De todos modos no desechamos estas opciones para futuros cursos.

A partir de los objetivos anteriormente expuestos, las actividades de aprendizaje se han clasificado en función de su tipología en las siguientes cuatro categorías:

- sesiones de teoría: los profesores presentan una serie de conceptos y técnicas, además de

ejemplos de su utilización. Los estudiantes deberán repasar fuera del aula los apuntes y sus notas para acabar de asimilar los contenidos. Las sesiones se realizan en grupos de entre 50 y 70 estudiantes.

- sesiones de ejercicios: los estudiantes deben resolver una serie de ejercicios de pequeño tamaño, poniendo en práctica los conceptos y técnicas presentados en las sesiones de teoría. Estas actividades se comienzan en aulas de ordenadores, y deben completarse fuera del aula. Las sesiones se realizan individualmente en grupos de entre 20 y 30 estudiantes.
- sesiones de prácticas: los estudiantes deben resolver unos problemas de un tamaño mayor que los ejercicios antes citados, de forma que deben decidir qué conceptos y técnicas han de utilizar en cada caso. Estas actividades se comienzan en aulas de ordenadores, y deben completarse fuera del aula. Las sesiones se realizan por parejas, en grupos de entre 20 y 30 estudiantes.
- ejercicios de autoevaluación: ejercicios para que el estudiante, al final de cada unidad didáctica, compruebe que ha asimilado los conceptos y técnicas presentados. Esta actividad la realiza el estudiante individualmente fuera del aula.

La figura 1 muestra el ciclo típico de una unidad didáctica. Cada unidad comienza con una sesión de teoría donde se presentan los fundamentos y ejemplos fundamentales; le sigue una sesión de ejercicios donde los alumnos resuelven ejercicios simples aplicando directamente los fundamentos explicados en la sesión de teoría; sesión de prácticas donde el alumno debe elegir los conceptos y técnicas que debe utilizar para resolver un problema más complejo. Después de una o varias iteraciones como ésta, la unidad didáctica siempre finaliza con un examen de autoevaluación, con el que el alumno puede

comprobar si el progreso ha sido el adecuado y puede afrontar el examen. La estructura varía ligeramente en el caso de la práctica final, ya que su desarrollo incluye varias sesiones seguidas.

Los 5 créditos LRU de la asignatura se han convertido en 4 créditos ECTS, es decir, una dedicación por parte del alumno de 100 horas de trabajo. De éstas, aproximadamente la mitad del trabajo se ha realizado en el aula y la mitad fuera. En total se han programado 10 sesiones de teoría, incluyendo dos sesiones finales para repaso y resolución de dudas (15 horas en el aula y 4 fuera), 7 de ejercicios (10,5 horas en el aula y 17,5 fuera), 12 sesiones de prácticas (24,5 horas en el aula y 24,5 fuera) y 4 autoevaluaciones (4 horas fuera del aula). Estos números varían ligeramente según el grupo en función de los días de fiesta.

No obstante, conviene aclarar que esta división entre trabajo en aula y fuera no se corresponde exactamente con la división entre trabajo con profesor y trabajo personal. Hay que recordar que un requisito fundamental de la experiencia fue el de no incrementar la dotación de profesorado. Además, la antigua división entre profesor de grupo de teoría (30 horas) y de prácticas (20 horas) igualmente tuvo que mantenerse. Así, mientras que la dedicación del profesor de prácticas no ha cambiado, las antiguas 20 sesiones de 1,5 horas del profesor de teoría, han pasado a ser 10 sesiones de teoría, 7 de ejercicios y 3 de prácticas. En las 10 sesiones de teoría se mantiene el trabajo con el grupo completo, pero en las 10 restantes de ejercicios y prácticas, el grupo se ha dividido en 2 ó 3 subgrupos (2 para un grupo y 3 para otro). Por tanto, en esas 10 sesiones de ejercicios y prácticas, el alumno tiene en total 10 ó 15 horas de profesor y el resto de trabajo personal. Así, el número global de horas con profesor está entre 30 y 35 sobre 100 horas globales de dedicación.

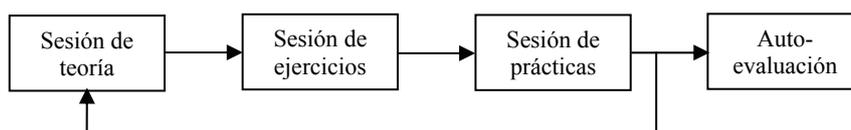


Figura 1. Ciclo de una unidad didáctica

### 3. Nuevo plan docente

La primera parte del diseño del plan docente es la definición de las competencias que el estudiante debe adquirir a través de la asignatura. Tal y como se ha indicado anteriormente, el objetivo fundamental de esta asignatura es que los alumnos adquieran las bases de la algorítmica y estructuras de datos, así como que sean capaces de desarrollar fluidamente programas de tamaño medio utilizando el lenguaje C.

Estos objetivos generales se concretan en la lista de competencias que se presenta a continuación. Por motivos de espacio no se describen en profundidad aquí pero pueden ser consultadas en el documento del plan docente disponible on-line [3]. En primer lugar se listan las competencias generales, que se refieren a habilidades no directamente relacionadas con la programación en sí, sino al ámbito profesional de un informático: capacidad de síntesis; capacidad de análisis; capacidad para aplicar el conocimiento en la práctica; e interés por la calidad. Y a continuación se incluyen las competencias específicas, que son las referidas a aspectos propios de la asignatura: capacidad para trabajar con un compilador y debugador; dominio de los tipos de datos estáticos básicos y compuestos; dominio de las estructuras de control; capacidad de resolución de problemas mediante diseño descendente y dominio de la utilización de funciones; dominio de los tipos de datos dinámicos y de la gestión dinámica de memoria; documentación y estructuración de código; y capacidad de lectura (rápida) de código en C.

A partir de estas competencias se han desarrollado una serie de actividades de aprendizaje para trabajar cada una de ellas. Estas actividades pueden englobarse en los cuatro tipos que han sido descritos anteriormente: sesiones de teoría, sesiones de ejercicios, sesiones de prácticas y ejercicios de autoevaluación.

En cuanto a la estructuración de los contenidos, ésta no ha sufrido variaciones importantes respecto de lo descrito en el apartado del contexto. Así, como resultado de cruzar los contenidos con las actividades de aprendizaje, hemos estructurado la asignatura en

cuatro unidades didácticas. Cada unidad didáctica tiene uno o varios ciclos de actividades como los que se han descrito anteriormente. A continuación se enumeran las unidades didácticas que se han definido:

Unidad didáctica 1. Primeros pasos: introducción, tipos de datos básicos y estructuras de control:

- Bloques de contenido: Conceptos generales (1), Tipos de datos básicos (2) y Expresiones, sentencias y estructuras de control (3)
- Actividades de aprendizaje: 3 sesiones de teoría, 1 sesión de ejercicios, 3 sesiones de prácticas y 1 sesión de ejercicios de auto-evaluación
- Dedicación total: 20 horas (12 en el aula, 8 fuera)

Unidad didáctica 2. Funciones y diseño descendente:

- Bloques de contenido: Descomposición funcional y diseño descendente (4)
- Actividades de aprendizaje: 1 sesión de teoría, 1 sesión de ejercicios, 1 sesión de prácticas y 1 sesión de ejercicios de auto-evaluación
- Dedicación total: 11 horas (5 en el aula, 6 fuera)

Unidad didáctica 3. Tipos de datos compuestos estáticos y declaración de tipos:

- Bloques de contenido: Tipos de datos compuestos estáticos (5) y Declaración de tipos propios (6)
- Actividades de aprendizaje: 2 sesiones de teoría, 3 sesión de ejercicios, práctica parcial y 1 sesión de ejercicios de auto-evaluación
- Dedicación total: 24 horas (11,5 en el aula, 16,5 fuera)

Unidad didáctica 4. Punteros y ficheros de texto:

- Bloques de contenido: Punteros, gestión dinámica de memoria y paso de parámetros por referencia (7) y Ficheros de texto (8)
- Actividades de aprendizaje: 2 sesiones de teoría, 2 sesión de ejercicios, 1 sesión de prácticas y práctica final y 1 sesión de ejercicios de auto-evaluación
- Dedicación total: 30 horas (15 en el aula, 15 fuera)

Como puede observarse, no se han incluido los detalles de cada actividad concreta. Éstas pueden ser muy variadas, en función de la competencia que se vaya a trabajar. Por ejemplo, se han planteado ejercicios específicos para trabajar la lectura de código o el diseño descendente.

En cuanto a la evaluación se ha mantenido la separación entre práctica (50% de la nota final de la asignatura) y lo que hemos denominado fundamentos teórico-prácticos (50% restante) y ambas deben ser aprobadas para superar la asignatura. Esta última parte se evalúa mediante dos exámenes, uno al finalizar la Unidad didáctica 2, durante la sexta semana, y otro al final el trimestre. El primero supone un 25% de la nota de esta parte y el segundo el 75%. Además, tal y como se ha indicado anteriormente, se han programado varias actividades de auto-evaluación, una al final de cada unidad didáctica, con el objetivo de que el estudiante vaya valorando su propio progreso. Así mismo, en las actividades de ejercicios el estudiante puede también evaluar su progreso, comparando sus soluciones con los resultados publicados. En todo caso, estas actividades de auto-evaluación no inciden en la nota.

En cuanto a la parte práctica, también se evalúa a partir de dos prácticas entregables, una a mitad de la asignatura (25% de la nota) y otra al final (75%). La práctica parcial tiene un peso del 25%, mientras que la final supone un 75%. Estas dos prácticas, y especialmente la final, permiten no sólo evaluar que el alumno haya adquirido las competencias específicas de la asignatura, sino también las competencias generales anteriormente mencionadas. Por otra parte, las otras prácticas que no inciden en la nota también servirán a los estudiantes para ir evaluando su propio progreso, comparando sus ejercicios con los resultados publicados.

Para acabar este apartado, hay que destacar que esta reorganización de la asignatura ha requerido la publicación de nuevos materiales docentes. Además de actualizar los apuntes de la asignatura, un documento de 70 páginas, se han preparado transparencias (unas 200) para agilizar las clases de teoría, colecciones de ejercicios resueltos, ejercicios de auto-evaluación, soluciones de prácticas, así como manuales de Linux y compiladores. Todos estos

materiales pueden consultarse en la web de la asignatura [3]. Allí también puede encontrarse el plan docente y una versión reducida de la programación temporal de la asignatura, especialmente enfocada a los alumnos.

#### 4. Estudio de la experiencia

En esta sección se describe el primer estudio llevado a cabo con la intención de evaluar la experiencia. En primer lugar se presentan resultados provenientes de tres fuentes: (1) el porcentaje de aprobados; (2) la valoración subjetiva, por parte de los alumnos, de determinados aspectos de los bloques de contenidos y tipos de actividades realizadas; y (3) la evaluación de la docencia por parte de los alumnos. Finalmente se incluye una discusión de todos los resultados presentados.

##### 4.1. Porcentaje de aprobados

El curso anterior, utilizando el antiguo plan docente, aprobaron 26 alumnos de un total de 77 presentados (un 33,8%). Este curso, con el nuevo plan docente (pero el mismo tipo de examen y prácticas que el curso pasado), han aprobado 47 alumnos de un total de 74 presentados (un 63,5%). Así, el porcentaje de aprobados se ha incrementado en un 87,9%. Además, como novedad, este curso no ha habido ningún alumno al que le haya quedado pendiente sólo el examen, lo que parece indicar que las prácticas han sido realmente útiles. En cambio el porcentaje de no presentados se ha mantenido en un 27%, cifra similar al curso anterior.

##### 4.2. Valoración de contenidos y actividades

Los resultados que se presentan en esta sección se basan en un cuestionario diseñado para evaluar la percepción subjetiva, por parte de los alumnos, de los siguientes aspectos:

- la dificultad de los diferentes bloques de contenidos
- la adecuación del número de horas asignadas a cada bloque de contenidos
- la adecuación del número de horas asignadas a cada tipo de actividad
- la utilidad de los diferentes tipos de actividad
- la dificultad de las prácticas parcial y final

Se utilizó una escala Likert [2] de 5 puntos para todas las cuestiones. Nótese que en las preguntas que tienen que ver con la adecuación de horas, los extremos corresponden a muy pocas (1) y demasiadas (5) horas, de modo que el valor ideal de adecuación del número de horas corresponde al 3.

Un total de 35 participantes (alumnos de la asignatura Programación I) rellenaron el cuestionario, de forma anónima, durante la última clase de teoría de la asignatura. De este modo, los alumnos ya habían casi completado la asignatura, y aún no habían sido evaluados. Con esto último pretendíamos evitar que las respuestas de los estudiantes fuesen influenciadas por sus resultados en el examen y la práctica final.

Para cada uno de los aspectos a estudiar, utilizamos la prueba de rangos con signo de Wilcoxon [4] para examinar diferencias individuales. Todas las pruebas fueron de dos colas.

### Resultados

Los participantes fueron consultados sobre la *dificultad* de los *contenidos* de cada una de las unidades didácticas: UD1 (tipos de datos básicos y estructuras de control), UD2 (descomposición funcional y diseño descendente), UD3 (tipos de datos compuestos) y UD4 (punteros, paso de parámetros por referencia y ficheros).

El orden en el que se listan los contenidos corresponde al orden en que éstos se imparten a lo largo del trimestre. Según el análisis estadístico, los alumnos encontraron cada bloque de contenidos significativamente más difícil que el inmediatamente anterior. Estas diferencias se ilustran en la figura 2.

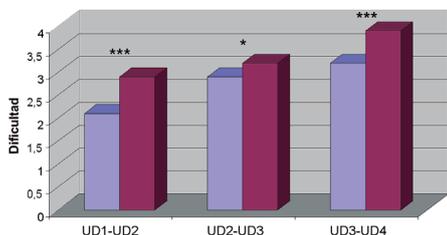


Figura 2. Diferencias significativas en la dificultad de los contenidos (\*  $P < 0,05$ , \*\*  $P < 0,01$ , \*\*\*  $P < 0,001$ )

En la figura, la altura de cada columna representa el valor promedio obtenido para la dificultad del bloque de contenidos correspondiente. En particular, observamos que los alumnos encuentran el bloque UD2 claramente más difícil que el UD1 ( $Z = 3,862$ ,  $P < 0,001$ ), el UD3 más difícil que el UD2 ( $Z = 2,358$ ,  $P < 0,05$ ) y el UD4 claramente más difícil que el UD3 ( $Z = 3,577$ ,  $P < 0,001$ ).

Analizando la percepción de los alumnos sobre la adecuación del *número de horas* dedicadas a cada bloque de *contenidos*, notamos que para todos los bloques se obtuvo un valor promedio menor que 3, es decir, menos horas de las adecuadas (véase figura 3). Además se observa que el valor medio para cada bloque de contenidos es menor que el valor medio para el bloque anterior. Analizando las diferencias significativas, encontramos que los alumnos piensan que la necesidad de aumentar el número de horas del bloque UD3 es mayor que la de aumentar el bloque UD1 ( $Z = 2,368$ ,  $P < 0,05$ ), y que la necesidad de aumentar el número de horas del bloque UD4 es claramente mayor que las de aumentar las de los bloques UD1 ( $Z = 3,840$ ,  $P < 0,001$ ) y UD2 ( $Z = 3,273$ ,  $P < 0,001$ ). Es decir, los bloques que parecen requerir un aumento más claro de horas son el UD3 y sobre todo el UD4.

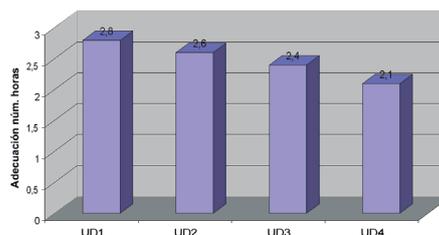


Figura 3. Valores promedio de la adecuación de horas dedicadas a cada bloque de contenidos

Los participantes fueron consultados sobre la *utilidad* de cada uno de los siguientes tipos de *actividad*: teoría (T), ejercicios (E), auto-evaluación (A), prácticas no evaluables (PNE), práctica parcial (PP) y práctica final (PF).

La figura 4 muestra los valores promedio obtenidos para cada uno de los tipos de actividad. Las diferencias significativas que

encontramos se muestran en la figura 5. En particular, los alumnos encontraron las sesiones de teoría menos útiles que las prácticas no evaluables ( $Z = 2,170$ ,  $P < 0,05$ ) y que la práctica parcial ( $Z = 2,002$ ,  $P < 0,05$ ), y claramente menos útiles que las sesiones de ejercicios ( $Z = 3,189$ ,  $P < 0,001$ ). Por otro lado, los alumnos encontraron las sesiones de ejercicios más útiles que las de auto-evaluación ( $Z = 2,627$ ,  $P < 0,01$ ).

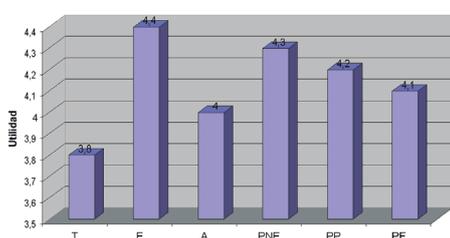


Figura 4. Valores promedio de la utilidad de los diferentes tipos de actividad

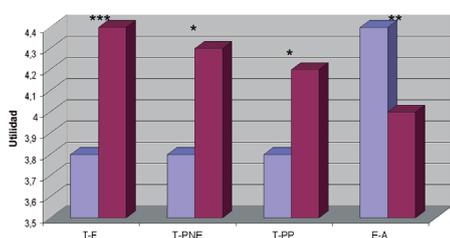


Figura 5. Diferencias significativas en la utilidad de los tipos de actividad (\*  $P < 0,05$ , \*\*  $P < 0,01$ , \*\*\*  $P < 0,001$ )

Analizando la percepción de los alumnos sobre la adecuación del número de horas dedicadas a cada tipo de actividad, notamos que se obtuvo un valor medio menor que 3 para todos los tipos de actividad excepto para las sesiones de teoría (véase figura 6). En este último caso el valor medio es ligeramente mayor que 3 (3.1). Es decir, los alumnos parecen pensar que sería necesario dedicar más horas a todos los tipos de actividad, exceptuando las sesiones de teoría. Sin embargo sólo encontramos dos diferencias significativas. En particular entre el número de horas dedicado a las sesiones de teoría en comparación con las dedicadas a las prácticas parcial ( $Z = 2,396$ ,  $P < 0,05$ ) y final ( $Z = 2,045$ ,  $P < 0,05$ ).

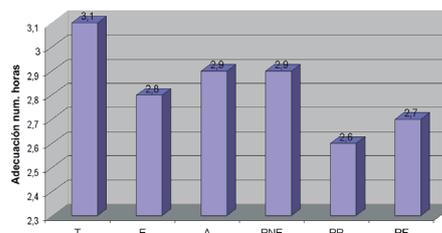


Figura 6. Valores promedio de la adecuación de horas dedicadas a cada tipo de actividad

Los participantes también fueron consultados sobre la dificultad de las prácticas parcial y final. Sin embargo, aunque el valor promedio obtenido para la práctica final (4,0) es mayor que el obtenido para la práctica parcial (3,8), no encontramos diferencia significativa.

### 4.3. Evaluación docente

En la UPF, la evaluación de la docencia se lleva a cabo por parte de los estudiantes, que al finalizar cada periodo docente (trimestre), valoran su satisfacción sobre determinados aspectos relacionados con la docencia y las asignaturas. En particular, cada alumno valora, en una escala de 0 a 10, las siguientes cuestiones: El profesor asiste a clase según el horario establecido; el profesor explica con claridad; se hace lo que prevé el programa de la asignatura; el material didáctico es adecuado; la asignatura es interesante; satisfacción global con la docencia recibida; las prácticas se han desarrollado satisfactoriamente.

Para cada asignatura-grupo-profesor se calcula la nota media en cada una de las preguntas. Para cada estudio y departamento se calcula el punto medio de todas las evaluaciones en cada una de las preguntas, y se establece una ordenación en percentiles (10%-35%-65%-90%) para cada estudio y departamento.

Hay que destacar que, si bien la evaluación docente de Programación I siempre había sido buena, ésta ha mejorado con el nuevo plan docente. La valoración en todas las preguntas ha sido muy superior a la media, situándose siempre en los percentiles 5 (90%-100%) ó 4 (65%-90%) de los estudios y el departamento.

#### 4.4. Discusión

Los alumnos han valorado claramente las sesiones de ejercicios como más útiles que las de teoría y auto-evaluación. Dado que uno de los principales cambios realizados con el nuevo plan de estudios ha sido convertir horas de teoría en horas de ejercicios, parece que, al menos desde la percepción de los estudiantes, este cambio ha sido un éxito.

Además, aunque de forma menos rotunda, las sesiones de teoría han sido valoradas como menos útiles que las de prácticas no evaluables y las de la práctica parcial. Es decir, parece que hay una tendencia a percibir como más útiles las sesiones prácticas que las teóricas. Puede extrañar entonces que las sesiones de la práctica final no aparezcan como más útiles que las de teoría. Quizá el motivo tenga que ver con el hecho de que los alumnos rellenaron el cuestionario antes de acabar dicha práctica.

Se podría pensar, en un primer momento, que si las sesiones de teoría son peor valoradas, puede deberse a una mala docencia por parte de los profesores. Sin embargo, hemos descartado esta opción, dado que la evaluación de la docencia de ambos profesores (por parte de los propios alumnos) ha sido excelente.

Por otro lado, la percepción de los alumnos sobre la dificultad de los bloques de contenidos se corresponde con lo que esperábamos. Cada bloque es considerado más difícil que el inmediatamente anterior.

En lo que respecta a la adecuación del número de horas, los estudiantes piensan que se debería dedicar más horas a todos y cada uno de los bloques de contenido, especialmente a los últimos. Y también creen que se debería dedicar más horas a todos los tipos de actividad, excepto a las sesiones de teoría. Las actividades para las que más horas adicionales requieren son las prácticas parcial y final. Esto nos hace pensar que aún podríamos reducir un poco el número de horas de teoría, y utilizar esas horas para sesiones supervisadas adicionales de prácticas parcial y final.

En todo caso, aunque creemos que el diseño del plan docente aún se puede mejorar de cara al curso que viene, el notable incremento en el porcentaje de aprobados avala el éxito de la experiencia.

#### 5. Conclusiones y previsiones para el próximo curso

En este artículo hemos descrito el nuevo diseño de la asignatura Programación I, en la línea de las directrices del EEES. Los objetivos que han guiado el diseño son: reducción del número de sesiones de teoría, reducción del salto entre teoría y práctica, incremento del trabajo efectivo de los alumnos, e incremento de la interacción entre profesores y alumnos. Además hemos detallado el primer estudio llevado a cabo con el fin de evaluar la experiencia. El estudio, que utiliza datos provenientes de tres fuentes diferentes, ha mostrado que la experiencia ha sido un éxito, y nos ha dado indicios sobre cómo mejorar aún más el diseño.

Tenemos la intención de estudiar la evolución de los alumnos que han cursado Programación I con el nuevo plan de estudios. Estudiaremos sus resultados en las asignaturas Programación II y Estructuras de Datos y de la Información I (ambas del segundo trimestre), comparándolos con los de estudiantes del curso pasado.

De cara al curso que viene, prevemos la inclusión de nuevas actividades de trabajo en grupo, así como la reducción (aún más) del número de horas de teoría.

#### Agradecimientos

Deseamos mostrar nuestro agradecimiento al equipo del PQE de la UPF, y en especial a Carme Hernández, por sus consejos y apoyo durante el diseño de esta experiencia.

#### Referencias

- [1] Costelloe, E. *Teaching Programming: The State of the Art*. The Center for Research in IT in Education. Dept. of Computer Science Education. Trinity College. Dublin. 2004.
- [2] Likert, R. *A Technique for the Measurement of Attitudes*. *Archives of Psychology*, 140, 1-55, 1932.
- [3] Navarrete, T., Ibáñez, J: web de la asignatura Programación I (2005/06). <http://www.tecn.upf.es/~tnavarrete/programacio1>
- [4] Wilcoxon, F. *Individual Comparisons by Ranking Methods*. *Biometrics*, 1, 80-83, 1945.