

Análisis y diseño de algoritmos

Organización curricular para un primer contacto con la disciplina

Francisco Palomo Lozano, Inmaculada Medina Bulo

Dpto. de Lenguajes y Sistemas Informáticos. Universidad de Cádiz.

Escuela Superior de Ingeniería de Cádiz. C/ Chile, s/n. 11003 Cádiz.

e-mail: {francisco.palomo, inmaculada.medina}@uca.es

Resumen

En este trabajo presentamos nuestra experiencia con una nueva organización curricular para la materia de análisis y diseño de algoritmos en los primeros cursos de los estudios de Informática y realizamos una propuesta docente para una asignatura introductoria.

1. Introducción

En el curso 2002–03 comenzó a impartirse en la Universidad de Cádiz (UCA) un nuevo plan de estudios de la Ingeniería Técnica en Informática de Gestión [13] con diversos cambios en la organización curricular que afectan al estudio de las estructuras de datos y los algoritmos. La nueva organización cuatrimestral dispone cuatro asignaturas para estas materias.

En el primer año, tras un curso de introducción a la programación (IP) y otro de matemática discreta (MD), se cubren las estructuras de datos lineales (ED I), a la vez que se afianzan los conocimientos en metodología y tecnología de la programación (MTP).

El segundo año simultanea el estudio de las estructuras de datos no lineales (ED II) con una primera asignatura de análisis y diseño de algoritmos (ADA I), que posee una segunda parte (ADA II) dedicada al estudio de las principales técnicas de diseño de algoritmos y a su análisis detallado.

Este enfoque, que reconoce el carácter independiente de la Algoritmia, está en consonancia con las últimas recomendaciones de la ACM/IEEE-CS [1], en las que el área AL ha pasado a estar dedicada en exclusiva al estudio

de los algoritmos y de la complejidad.

	1 ^{er} Cuatrimestre		2 ^o Cuatrimestre	
	Asig.	Ctos.	Asig.	Ctos.
1 ^o	IP	3T+4,5P	MTP	3T+3P
	MD	3T+3P	ED I	3T+3P
2 ^o	ED II	3T+3P		
	ADA I	3T+1,5P	ADA II	3T+1,5P

Cuadro 1: Ubicación de las asignaturas.

Algunos de los principios que inspiran esta organización son:

1. El estudio de los algoritmos puede separarse y superponerse al de las estructuras de datos, si bien los enfoques han de ser complementarios y coordinados.

Hay estructuras de datos y algoritmos que constituyen «conceptos recurrentes» en ambos tipos de asignaturas, pero que se estudian bajo enfoques distintos.

2. Se deben conocer los rudimentos del análisis de algoritmos para apreciar la bondad de las técnicas de diseño.

Si la eficiencia ha de ser una guía de diseño, es imprescindible conocer lo suficiente sobre análisis como para poder intuir a priori si un diseño va a ser o no adecuado.

De acuerdo con estos principios, debe existir una primera asignatura centrada en el análisis, pero que exponga buenos principios de diseño, que requiera algunos conocimientos de estructuras de datos, pero no todos, y que cimente el posterior estudio de las técnicas de diseño.

Es esta asignatura, ADA I, a la que prestaremos nuestra atención en este trabajo.

2. Asignatura de ADA I

Los objetivos fundamentales que pretende cubrir ADA I se resumen en aprender a:

1. Analizar formalmente la complejidad espacial y temporal de algoritmos básicos.
2. Programar en el laboratorio los algoritmos explicados en clase y contrastar los resultados experimentales obtenidos con la teoría expuesta.

Como se aprecia en el cuadro 1, la asignatura posee en el plan de estudios 3 créditos teóricos y 1,5 prácticos.¹

Por ello, no tratamos de cubrir mucha materia superficialmente, sino poca en profundidad, dedicando a los aspectos fundamentales y a la resolución de problemas en clase todo el tiempo posible.

El cometido de la asignatura es, pues, sentar una sólida base para el estudio de ADA II, que puede entonces dedicarse a exponer las técnicas de diseño de algoritmos, como «divide y vencerás», algoritmos devoradores y programación dinámica, así como a analizar en detalle ejemplos notables de cada una de ellas.

Los prerrequisitos se encuentran repartidos entre diversas asignaturas. Por un lado, se esperan una serie de conocimientos matemáticos: que el alumno sepa resolver ecuaciones de recurrencia lineales, acotar un sumatorio, emplear el principio de inducción, etc.

Por otro, el alumno debe estar ya familiarizado con el diseño iterativo y recursivo elemental, la abstracción, la modularidad y el empleo de precondiciones, postcondiciones e invariantes. Las estructuras de datos lineales son también necesarias desde los primeros algoritmos.

Por último, cierto dominio del lenguaje C, empleado en primero, es imprescindible para comenzar las prácticas. Utilizaremos un subconjunto apropiado de C++ siguiendo el paradigma de la programación genérica, que nos permite separar con claridad las estructuras de datos de los algoritmos que las manipulan.

¹Esto corresponde a 30 horas de teoría y 15 de prácticas, pero no es extraño que el calendario lectivo fuerce un número de horas algo inferior.

2.1. Contenidos teóricos

Los contenidos teóricos se organizan como sigue. Téngase en cuenta que el tiempo indicado junto a cada tema incluye el que se dedica a la resolución de problemas en clase, que suele ser de un par de horas por tema.

T1 Introducción y conceptos básicos (4 h)

Problemas, algoritmos y programas. Especificación. Corrección y eficiencia. Ejemplo: multiplicación rusa. Principio de invariancia. Definición informal de orden.

T2 Órdenes asintóticos (4 h)

Órdenes asintóticos. Operaciones.

T3 Complejidad algorítmica (8 h)

Tiempo y espacio. Enfoques en el análisis de algoritmos. Peor caso, mejor caso y promedio. Análisis de las estructuras de control. Ejemplo: algoritmos elementales.

T4 Algoritmos clásicos y su análisis (8 h)

Búsqueda. Métodos directos de ordenación. Ordenación por montículo. Componentes conexas.

T5 Complejidad de los problemas (6 h)

Problemas tratables e intratables. Problemas y árboles de decisión. Cotas inferiores de complejidad. Reducibilidad. Clases de complejidad.

El tema de introducción sirve de enlace con las asignaturas de primero donde el alumno ha estudiado ya superficialmente las nociones de corrección y eficiencia. También permite fijar la notación a emplear y presenta un primer ejemplo con el que se ilustran sistemáticamente todos los conceptos expuestos.

Se elige la multiplicación rusa como primer ejemplo porque sorprende a los alumnos de varias formas: refuta la muy extendida idea de que no hay otra forma de multiplicar que la que aprendimos de pequeños, es uno de los algoritmos documentados más antiguos [2] que se conocen y supone la implementación más simple que se puede realizar de la multiplicación en una computadora digital binaria.

El segundo tema expone con rigor los órdenes O , Ω y Θ y sus operaciones. A diferencia de lo que se hace en muchos textos, se definen los órdenes como conjuntos de funciones, como se predica en [3] y se practica en [4, 12].

Se insiste en las propiedades formales de los órdenes, ya que nos permitirán luego razonar con ellos mecánicamente, con seguridad, en vez de tener que recurrir continuamente a la intuición, como desgraciadamente ocurre en gran número de textos de esta disciplina.

El tercer tema comienza con las nociones de espacio y tiempo algorítmicos, abstractos por naturaleza, y su relación con sus homónimas reales. Tras justificar el análisis por el método de la operación crítica, se expone la necesidad de clasificar los ejemplares según su tamaño y los casos de análisis que surgen al hacerlo.

Por su dificultad intrínseca, prestamos especial atención al análisis en el caso promedio. Los ejemplos que se utilizan son sencillos (cálculo del mínimo, inserción en orden y comprobación de ordenación) y sirven de base a los algoritmos del siguiente tema.

El cuarto tema cubre la búsqueda secuencial y los métodos directos de ordenación (intercambio directo y selección e inserción directas), e incluye el análisis en el promedio de la búsqueda y de la ordenación por inserción.²

Nótese que hasta aquí sólo se han necesitado las estructuras lineales de ED I. El tema prosigue con algoritmos más elaborados que requieren algunas estructuras no lineales ya impartidas en ED II a estas alturas del curso.

El primero es un método de ordenación no directo: el algoritmo de Williams u ordenación por montículo. Se analizan con todo rigor dos algoritmos de construcción de montículos, el clásico de Williams y el lineal de Floyd. Se concluye con un algoritmo de cálculo de componentes conexas mediante estructuras de partición, durante cuyo análisis se incide en el impacto que la elección de la estructura de datos subyacente tiene sobre el algoritmo.

²La búsqueda binaria, la ordenación por fusión y la de Hoare se reservan a ADA II, ya que forman el canon de las distintas subespecies de algoritmos de «divide y vencerás» [4] y su análisis riguroso se aborda mejor con las técnicas que allí se explican.

El último tema es una introducción estándar a la teoría de la complejidad de los problemas, donde se intenta que el alumno comprenda la diferencia entre la complejidad de un algoritmo y la de un problema, y la existencia de una dificultad intrínseca a algunos problemas, insalvable mediante un aumento lineal de la potencia de cálculo disponible.

2.2. Contenidos prácticos

Las prácticas se organizan en varias sesiones de una hora de duración siguiendo un modelo de laboratorio cerrado en el que se proporciona al alumno un guión que contiene tanto las explicaciones como los ejercicios a desarrollar.

P1 Programación genérica con C++ (5 h)

C++ como un C mejorado. Programación genérica en C++. Biblioteca STL.

P2 Generación de ejemplares de prueba (2 h)

Pruebas de caja negra. Números pseudoaleatorios. Permutaciones pseudoaleatorias.

P3 Medida del tiempo de ejecución (4 h)

Tipos de medida. Formas de medir el tiempo. Factores que influyen en la medida. Elección de los ejemplares de prueba. Ejemplo: números de Fibonacci.

P4 Comparación entre algoritmos de ordenación (4 h)

Métodos directos de ordenación por comparación. Algoritmos de ordenación de la biblioteca del lenguaje.

El enfoque seguido en las prácticas, fundamentado en una combinación de programación genérica con C++/STL, empleo de herramientas de software libre y técnicas experimentales de análisis, ha sido objeto de un artículo específico [9] al que remitimos al lector interesado.

3. Materiales didácticos

Aparte de las habituales transparencias y boletines de problemas, hemos introducido algunos materiales didácticos que estimamos merecen algunos comentarios.

3.1. Curiosidades

Intentamos motivar al alumno proporcionándole toda suerte de referencias históricas y curiosidades, animándole a buscar información adicional en Internet.

Por ejemplo, al estudiar la multiplicación rusa se les habla del papiro de Ahmes [2]. Al hablar de la sucesión de Fibonacci, del canon de belleza griego, de Leonardo da Vinci y su Hombre de Vitrubio, y de la constante aparición del número áureo en la naturaleza.

3.2. Breves repasos

No entraremos en las causas, diversas y complejas, por las que un alumno llega a un segundo curso universitario de las características que nos ocupan sin la formación requerida. Simplemente constatamos la existencia del problema, ante el que tenemos tres opciones:

1. Rechazar el problema. Al fin y al cabo, no es nuestra culpa. Sigamos con el plan docente preestablecido y dejemos bien claro que éste no es nuestro problema.
2. Asumir el problema como nuestro. Al fin y al cabo, puede que el alumno no tenga la culpa. Expliquemos en clase todo aquello que no sepa aun a costa de nuestro plan docente.
3. Asumir el problema solidariamente. Proporcionemos los medios necesarios para que el alumno con su esfuerzo personal y algo de nuestra ayuda supla sus carencias, pero sin sacrificar el plan docente.

Es en este tercer estadio en el que nos encontramos actualmente. Un elemento indispensable para llevarlo a cabo es el «breve repaso».

Un breve repaso es un documento corto,³ autocontenido, que un alumno puede estudiar holgadamente en una tarde por sí mismo y que ha de servirle para eliminar una laguna muy concreta de su formación sobre un prerrequisito de la asignatura.

Actualmente se han elaborado y puesto a disposición de los alumnos los siguientes:

1. Verificación de algoritmos: corrección, lógica de Floyd-Hoare, verificación a posteriori de algoritmos iterativos.
2. Recursividad lineal: recursividad lineal y final, relación con la iteración, generalización, desplegado y plegado.
3. Acotación de un sumatorio por una integral: técnica de acotación integral, interpretación gráfica, aplicación al cálculo de órdenes asintóticos.
4. Ecuaciones de recurrencia: relación con el cálculo de sumatorios, clasificación, resolución por el método de la ecuación característica, etc.

Los breves repasos se suelen acompañar de una abundante bibliografía de consulta que, sin ser de obligada lectura para su comprensión, ha de servir al alumno interesado para ampliar sus conocimientos.

3.3. Vídeos

University Video Communications (UVC) es una empresa que centra su actividad fundamentalmente en el ámbito académico y que distribuye conferencias de personas de reconocido prestigio internacional en la universidad o en la industria.

Las conferencias son normalmente divulgativas y de una gran calidad técnica, pero se encuentran normalmente en inglés, sin subtítular, lo que suele suponer un inconveniente para el alumno medio, con escasos conocimientos de lenguas extranjeras.

El último tema de teoría, dedicado a la complejidad de los problemas, se ha impartido utilizando un vídeo con una conferencia del Prof. Richard Karp [7], uno de los padres de la teoría de la complejidad, que lleva por título «NP-complete problems». En ella se cubre magistralmente una parte importante de los conocimientos generales que se espera posea un alumno sobre complejidad al terminar la asignatura.

Para romper la barrera del idioma los profesores transcribieron la conferencia en su idioma original, entregando la transcripción a los

³Idealmente, de menos de 10 páginas.

alumnos,⁴ previamente a la emisión de la conferencia.

3.4. Guiones de prácticas y software libre

Las prácticas de laboratorio son cerradas, por lo que el guión de prácticas juega un papel fundamental para el alumno. No sólo contiene los ejercicios a realizar en cada práctica, sino que, a modo de apuntes, contiene explicaciones sobre el trabajo a desarrollar o sobre técnicas de programación o funciones de biblioteca que pueden ser útiles en su realización.

Pensamos que esto no exime en ningún modo al profesor de dedicar algunos minutos al comienzo de la clase a orientar a los alumnos sobre el desarrollo de sus tareas y las dificultades que puedan encontrarse.

Por otro lado, intentamos promover en nuestros alumnos el empleo de software libre, proporcionando información sobre las herramientas disponibles y un entorno adecuado para la realización de las prácticas en el laboratorio. Utilizamos un sistema operativo GNU/LINUX, la última versión estable del compilador GNU C++ y algunas herramientas más, principalmente:

- GNU PLOT, que permite automatizar determinadas tareas de análisis y representación gráfica, ya que se programa de manera sencilla y permite obtener gráficas de calidad en una gran variedad de formatos.
- GNU MAKE, que está especialmente diseñada para ayudar a rehacer trabajos monótonos que deban repetirse cada vez que ciertos ficheros sean modificados. Ésta es justo la situación que nos encontramos en muchas ocasiones en el laboratorio, cuando hay que modificar el código de un experimento, recompilarlo, obtener los nuevos tiempos y redibujar las gráficas.

Para ahorrar trabajo al alumno, suministramos unos esqueletos tipo de los ficheros de entrada para las herramientas anteriores, permitiendo así que se concentre en el código del experimento en sí, y no en su organización.

⁴Deliberadamente sin acompañarla de una traducción a nuestra lengua, ya que creemos que es importante que empleen recursos en lengua inglesa.

4. Campus virtual

Ya que nuestra universidad ha adoptado institucionalmente la plataforma comercial de enseñanza virtual WebCT (www.webct.com), los autores decidieron incorporar su asignatura al programa de tutorías electrónicas y docencia en la red promovido por el Vicerrectorado de Ordenación Académica de la UCA.

Sin ser la plataforma ideal para el docente, que ha de pelear con aspectos bastante idiosincrásicos de la misma, hemos de reconocer que es cómoda de utilizar para el alumno y que, en general, ha sido muy bien aceptada.

Los datos de los alumnos se cargan de manera centralizada a través de los disponibles en secretaría liberándonos así de una de las tareas de administración más pesadas. Cada alumno recibe automáticamente un identificador de usuario y una contraseña común a todos los cursos en los que está matriculado.

Seleccionamos y pusimos a disposición de los alumnos varias herramientas de la plataforma:

- Agenda (recordatorios de fechas, cambios de última hora, etc.)
- Herramientas de organización de contenidos (plan docente, apuntes, transparencias, problemas, guiones, listados, etc.)
- Herramientas de comunicación (foros de debate, correo interno y charla en línea)

Sin duda, las herramientas más utilizadas han sido las de comunicación. Curiosamente, hemos constatado que no se emplea en absoluto la charla en línea. Puede que los alumnos deseen charlar en línea a través de sus propios canales, lejos de la mirada del profesor.

En la actualidad mantenemos varios foros: uno principal para dudas generales acerca de la organización docente, dos de contenidos para dudas relacionadas con la materia teórica y la práctica, respectivamente, otro de anuncios importantes de obligada lectura, otro privado para los profesores de la asignatura y, por último, uno de sugerencias.

Inicialmente, manteníamos un único foro para los contenidos teórico-prácticos, pero algunos alumnos expresaron su desconocimiento

acerca de cuestiones aclaradas a otros compañeros en él por contener éste un gran número de mensajes y estimar que carecían del tiempo necesario para buscar lo que les interesaba. Se optó, pues, por dividir el foro de contenidos en dos para facilitar su empleo.

Estimamos que el foro es, probablemente, la herramienta de comunicación más útil ya que permite que los alumnos pierdan el miedo a contestar a las dudas de sus propios compañeros y facilita que el profesor pueda detectar y responder de una vez dudas generales o aclarar conceptos mal aprendidos.

Los profesores de la asignatura mantienen, no obstante, dos horas semanales de tutorías presenciales aparte de atender las tutorías electrónicas.

No obstante, hemos constatado una disminución muy preocupante del habitualmente escaso número de alumnos que asisten a las tutorías presenciales. Parece ser que los alumnos se sienten realmente cómodos con el sistema electrónico de tutorías, lo que nos alegra y nos preocupa a la vez, ya que estimamos que hay dudas y cuestiones que es mejor resolver «cara a cara» con el profesor.

Una herramienta disponible, pero que no hemos utilizado, es la que permite realizar pruebas tipo «test». Esto se debe principalmente a la dificultad que presenta una asignatura como la que nos ocupa, basada fundamentalmente en la resolución de problemas, para establecer pruebas de este tipo que sean de utilidad.

Actualmente, el correo electrónico interno se emplea exclusivamente para la resolución de dudas en privado. Desde que se implantó este sistema, en cada curso se ha contabilizado una media de 192 comunicaciones por esta vía, un número algo inferior a los 254 mensajes gestionados en los distintos foros. Como puede inferirse fácilmente, la gestión de tal cantidad de información (una media de 30 comunicaciones semanales) supone un esfuerzo añadido para los profesores implicados.

5. Método de evaluación

Por razones relacionadas con la implantación gradual del nuevo plan de estudios y la resis-

tencia natural que presentan los alumnos de planes antiguos a la adaptación a un nuevo plan, al menos durante sus primeros años, el primer curso académico del nuevo plan contamos con un número inusualmente bajo de alumnos en clase (inferior a los 75) para los estándares de nuestra titulación.⁵

Esto nos permitió formar tres grupos prácticos, en torno a los 25 alumnos cada uno, y llevar a cabo una evaluación continua de la parte práctica de la asignatura, no sin un considerable esfuerzo por parte de los profesores implicados.

Para cada práctica, el profesor corregía los errores de programación, concepto e, incluso, estilo. El alumno acudía a una entrevista personal donde se le explican los diversos problemas encontrados y se abre un nuevo plazo para su modificación, entrega y corrección final. Desgraciadamente, el aumento del número de alumnos ha tornado este sistema inviable, por lo que hemos vuelto a un sistema de evaluación tradicional con la obligatoriedad de la entrega final de algunas prácticas seleccionadas.

En cualquier caso, hemos observado que los alumnos apenas leen los guiones de prácticas y que, más bien, tienden a comenzar directamente con la resolución de los ejercicios, en ocasiones sin comprender bien qué es lo que están haciendo o qué se les pide realmente. Encontramos difícil combatir esta tendencia, probablemente, acumulada a lo largo de años.

Para la evaluación de la parte teórica se ha realizado un examen final. El examen final consta principalmente de problemas en los que el alumno tiene que aplicar reiteradamente los conocimientos adquiridos en todo el temario. Esto, que es posible gracias al escaso número de temas, es una forma de evitar que se dejen de estudiar algunos de ellos en la esperanza estadística de que no aparezcan en el examen.

Los alumnos con notas superiores a 4 en el examen teórico, aprobado en prácticas y participación activa en las clases y en el campus virtual obtienen el aprobado global de la asignatura.

⁵La antigua asignatura «Metodología y Tecnología de la Programación II», donde se impartía materia relacionada, llegó a tener 280 alumnos.

6. Evaluación de la propuesta

Los autores participan en un proyecto de formación continua integrado en el Plan Andaluz de Formación del Profesorado Universitario de la Unidad para la Calidad de las Universidades Andaluzas (UCUA).

Este proyecto de formación, de duración anual, se estructuró en su primera edición sobre tres ejes fundamentales:

1. Alumnos: motivación, relaciones alumno-profesor, técnicas de diálogo, y obtención de información sobre su perfil.
2. Actividad docente: exposición en el aula, materiales de estudio, recursos docentes y métodos de evaluación.
3. Planificación docente: coordinación entre asignaturas, revisión y definición del ámbito de conocimientos de cada una.

Además, se consideró el estudio de tres categorías más, de naturaleza transversal, ya que afectan a las señaladas anteriormente: el papel de las TIC en la educación, la adaptación e impacto de los créditos ECTS y los principios éticos que deben regir la actividad del docente, del alumno y del futuro profesional.

Como parte del proyecto, se desarrolló una encuesta exhaustiva (alrededor del centenar de preguntas) sobre diferentes aspectos relacionados con las asignaturas. Pasamos a exponer algunos resultados obtenidos sobre ADA I.

El 70 % de los alumnos encuestados tiene dos o más asignaturas pendientes de primer curso. La tasa de abandono es muy alta: la mitad de los alumnos abandonan la asignatura, indicando como principal causa su dificultad.

En este sentido, el 95 % de los alumnos piensa que la asignatura es difícil o muy difícil, aunque el 65 % piensa aprobarla dentro del curso académico. Sin embargo, y sorprendentemente, más de la mitad afirma que se matricularía si la asignatura no fuera obligatoria. Esto puede deberse al hecho de que el 81 % la consideran interesante. Los alumnos que no esperan aprobar la asignatura reconocen como causas principales su falta de base o el estar cursando demasiadas asignaturas simultáneamente.

Por último, queremos destacar que el 89 % de los encuestados opina que la implantación del campus virtual ha sido muy positiva para el aprendizaje y seguimiento de la asignatura.

No obstante, hemos de advertir que estos datos han de ser interpretados con cierta cautela, ya que la participación en el proceso fue voluntaria y, por consiguiente, no muy alta, estando en torno al 35 % de los matriculados.

7. Conclusiones

Se ha descrito y evaluado una propuesta docente para una asignatura introductoria de análisis y diseño de algoritmos, una disciplina que forma parte de la materia troncal en los planes de estudio de Informática y considerada fundamental en las recomendaciones curriculares internacionales.

Se han introducido algunos recursos didácticos como los breves repases, el empleo de vídeos con conferencias en versión original, las prácticas estructuradas en guiones con empleo exclusivo de software libre y el campus virtual.

La evaluación de la propuesta, las opiniones de los alumnos y nuestra experiencia durante el curso nos han hecho reflexionar sobre algunas de las posibles mejoras a introducir en sucesivos cursos, que pasamos a describir.

Estimamos que el análisis de las discusiones que se producen en los foros es una fuente magnífica para recabar información acerca de los verdaderos puntos débiles de la formación de nuestros estudiantes. Esta información nos está permitiendo elaborar:

- La lista de preguntas más frecuentes de la asignatura, tanto para la parte teórica como para la práctica.
- Breves repases de prerrequisitos específicos en los que detectamos lagunas en la formación del alumno.

Creemos que distribuir algunos de los exámenes resueltos por el profesor (pero no todos) haciendo énfasis en el análisis de su enunciado y en el estilo de presentación de la solución para cada problema tipo, ayuda a mejorar los resultados obtenidos.

Estamos convencidos de que la experiencia con el empleo de los vídeos de UVC ha sido tremendamente enriquecedora. Para próximos cursos, tenemos pensado incluir en el primer tema una conferencia del Prof. E. W. Dijkstra [5] titulada «Reasoning about programs» y disponible en la misma colección. Sería interesante también completar las transcripciones de las conferencias con glosarios, aclaraciones, preguntas de comprensión y otros recursos tendentes a aumentar su valor didáctico.

En esta línea, valoramos también la posibilidad de utilizar algunos de los materiales audiovisuales disponibles a través del CD-ROM que acompaña a [10], una obra no muy conocida en nuestro país, pero de indudables cualidades pedagógicas.

Consideramos también interesante recomendar a nuestros alumnos otras obras que no gozan de gran difusión aún, pero que, a buen seguro, pronto lo harán.

A. Levitin propone en [8] una nueva taxonomía para el estudio de las técnicas de diseño de algoritmos, que adereza con numerosos juegos de ingenio y un estilo expositivo claro, riguroso y muy cuidado.

En [6], los autores proponen problemas ligados a aplicaciones reales que no suelen encontrarse en otros textos, así como numerosas notas y ejercicios destinados a estimular la curiosidad del lector y despertar en éste el deseo de buscar recursos en Internet como los que los autores han recopilado para la obra.

Por último, pensamos que [11] puede servir para aumentar el interés de nuestros alumnos por la programación en general. Resulta atractiva la posibilidad de asignar trabajos de programación sobre problemas que ya han sido cuidadosamente preparados y clasificados según su dificultad y popularidad. Por supuesto, muchos de estos problemas requieren estructuras de datos y algoritmos ingeniosos. La entrega, seguimiento y corrección de los trabajos puede realizarse a través de Internet con el mismo sistema que se utiliza en las competiciones internacionales de programación, que emplea «jueces robot». De hecho, esta obra sirve de manual de entrenamiento para la participación en dichos eventos.

Referencias

- [1] ACM/IEEE. Joint task force on computing curricula. *Computing curricula 2001*. ACM Press y IEEE Computer Society Press (2001)
- [2] Ahmes. *Reglas para obtener el conocimiento de todo lo oculto*. Papiro «Rhind», Museo Británico (1700 A.C.)
- [3] Brassard, G. *Crusade for a better notation*. ACM Sigact News, 17(1) (1985)
- [4] Brassard, G. y Bratley, T. *Fundamentos de Algoritmia*. Prentice-Hall (1997)
- [5] Dijkstra, E. W. *Reasoning about programs*. Stanford University Distinguished Lecture Series. University Video Communications (1991)
- [6] Johnsonbaugh, R. y Schaefer, M. *Algorithms*. Pearson (2004)
- [7] Karp, R. *NP-complete problems*. Stanford University Distinguished Lecture Series. University Video Communications (1993)
- [8] Levitin, A. *Introduction to the design & analysis of algorithms*. Addison-Wesley (2003)
- [9] Palomo Lozano, F. y Medina Bulo, I. *Análisis híbrido: una propuesta práctica*. Actas de las IX Jornadas de Enseñanza Universitaria de la Informática. Thomson (2003)
- [10] Skiena, S. S. *The algorithm design manual*. Telos (1998)
- [11] Skiena, S. S. y Revilla, M. A. *Programming challenges. The programming contest training manual*. Springer (2003)
- [12] Peña Mari, R. *Diseño de programas: formalismo y abstracción*. Prentice-Hall, 3ª ed. (2004)
- [13] Universidad de Cádiz. *Plan de estudios conducentes al título de Ingeniero Técnico en Informática de Gestión*. BOE N° 171 de 18 de julio de 2002 (2002)