

Enseñanza por descubrimiento de los algoritmos de ordenación

Agustín Cernuda del Río
Departamento de Informática
Universidad de Oviedo
guti@uniovi.es

Resumen

La enseñanza de algoritmos tropieza frecuentemente con diversos problemas de desinterés o comprensión deficiente por parte del alumno. La metodología didáctica de lección magistral pura, tan habitual en el sistema universitario, no contribuye a evitar tales problemas.

Es posible adoptar algunas variantes metodológicas para la exposición en el aula de determinados algoritmos. En concreto, cabe plantear estrategias semi-participativas de descubrimiento, una basada en el error (para el algoritmo de ordenación por burbuja) y otra de análisis y síntesis (para el de ordenación por inserción directa). En este artículo se describe la aplicación de ambas propuestas.

1. Introducción

1.1. Entorno académico

Lo aquí descrito se llevó a cabo en la Escuela Universitaria de Ingeniería Técnica en Informática de Oviedo (EUITIO), en la asignatura Metodología de la Programación (MP) [2], cuatrimestral (2º cuatrimestre), troncal de primer curso [4] en dos titulaciones (Gestión y Sistemas). Es la segunda dedicada a la programación en primero, muy importante para el posterior desempeño en otras asignaturas.

Es una asignatura de adquisición de habilidades y de notable dificultad; el porcentaje de aprobados sobre presentados ronda el 50%, pero con índices altos de absentismo y abandono.

Los objetivos de esta asignatura cubren, en sus aspectos técnicos, fundamentos de programación y algorítmica, y también aspectos de metodología (análisis, diseño, notación, pruebas...) En la primera área citada se incluyen unidades didácticas sobre estructuras de datos y algoritmos básicos; esto incluye algoritmos de búsqueda (secuencial y dicotómica) y de ordenación (burbuja con señal e inserción directa).

1.2. Aspectos docentes

En cursos precedentes en esta y otras asignaturas se había observado que muchos alumnos, en sus implementaciones de algoritmos de ordenación, solían incurrir en ciertos errores típicos y previsibles. Frecuentemente ni siquiera habían llegado a una comprensión real de los algoritmos, y mucho menos a una asimilación efectiva.

Se venía empleando una técnica expositiva, con algunas mejoras:

- Medios visuales (incluyendo presentaciones con animaciones).
- Poner en relación los algoritmos de ordenación con los de búsqueda (que los alumnos comprendían con mayor facilidad).
- Sugerir recursos de estudio autónomo (como diversos programas de simulación).

Ante las deficiencias de comprensión nos planteamos probar otra metodología docente. Además, la mejor forma de combatir el absentismo y el abandono, según estudios aplicables a nuestro entorno [5], es precisamente la mejora en la metodología docente.

1.3. Objetivos

Lo que se juzgó conveniente mejorar [1], y por tanto se planteó como objetivo a la hora de seleccionar la nueva estrategia, fue:

- Posición más activa del alumno para el aprendizaje [7].
- Mejorar la experiencia del alumno en clase (mayor interés, atención y motivación).
- Aproximación más intuitiva a los algoritmos.
- Relacionar los contenidos con conocimientos o experiencia personal del alumno.

En especial, parecía interesante poner de manifiesto que los algoritmos se podían entender, en realidad, basándose en conocimientos propios, de forma constructiva y no memorística.

2. Ordenación por burbuja con señal

2.1. Fundamentos de la estrategia docente

Se imparte la variante de burbuja con señal porque parece más intuitiva para el alumno. La variante de bucles *for* anidados da una solución al problema, pero al alumno le cuesta entender por qué el algoritmo funciona. En la burbuja con señal, basta con entender el mecanismo de una pasada, y luego se hace patente que el algoritmo realiza pasadas hasta que ya no sea necesario; además, da pie a introducir algunas discusiones sencillas sobre eficiencia (cómo el algoritmo acaba enseguida si el vector está inicialmente ordenado, y cómo siempre da una pasada más de lo estrictamente necesario).

Los fallos de comprensión más frecuentes son:

- Errores en el intercambio de valores entre dos posiciones del vector. Aunque sea un algoritmo elemental, los alumnos tardan en asimilarlo completamente, y al estudiar la ordenación muchos aún no lo han hecho.
- Deficiente comprensión de que en cada paso del bucle el algoritmo opera con una pareja de elementos contiguos y no desplaza ningún valor una distancia mayor que uno.
- Confusión entre los pasos del bucle *for* y las pasadas completas del bucle; muchos alumnos creen que cuando el bucle *for* termina una pasada el algoritmo ya ha terminado.
- Llevar la variable del bucle *for* al final del vector, como de costumbre, cuando en este caso debe detenerse en el penúltimo elemento.
- Deficiente comprensión del papel de la variable booleana (señal) y, por extensión, de la relación entre el bucle *for* y el *while*.
- En el caso de vectores numéricos, un problema recurrente en alumnos novatos es la confusión entre los índices y los elementos.

Algunos de los problemas se corresponden con limitaciones del algoritmo en comparación con el esquema mental del alumno. Cualquier alumno puede ordenar manualmente una secuencia o intercambiar elementos; el problema es que estará aplicando intrínsecamente capacidades que un programa de ordenador no tiene. Cuando se traslada el problema de la ordenación a un

algoritmo, el alumno espera inconscientemente que algunas de esas capacidades actúen.

Existen diversas formas de denominar lo que aquí se plantea para promover un aprendizaje activo del alumno. Puede decirse que abordaremos una estrategia didáctica creativa ([8], p.122), o que sustituiremos un método expositivo por un método de indagación [3]. En terminología de Rodríguez González [6] se trata de una estrategia de aprendizaje como elaboración o construcción de conocimientos, aunque debido a la especificidad del problema no quepa hablar de un entorno abierto o de resolución de problemas.

No habrá exposición previa del algoritmo ni se mostrará código alguno. El alumno simulará directamente en clase el algoritmo manualmente (no escribiendo un programa), pero de manera que se le impida sacar partido de capacidades que el algoritmo no tiene. El algoritmo de ordenación por burbuja con señal es una solución relativamente natural e intuitiva, una vez que se comprenden las restricciones del funcionamiento de una máquina de cómputo. Si colocamos al alumno en una situación en la que él mismo esté sometido a tales restricciones, es muy probable que acabe *inventando* el algoritmo como única salida para resolver el problema.

En particular, se persigue que el alumno cometa algunos de los errores típicos mencionados, de modo que los identifique por sí mismo como callejones sin salida. El acto de identificar el error e interiorizar la forma de evitarlo en el intento siguiente puede proporcionar un aprendizaje significativo ([8], pág. 211).

2.2. Diseño de la situación de aprendizaje

Podemos pensar en que el profesor limite las acciones del alumno en el proceso de ordenación. Sin embargo, no es una estrategia óptima, ya que:

- Inevitablemente están presentes las implicaciones de la relación asimétrica entre profesor y alumno, y se hace patente la figura del profesor que *corrige* al alumno.
- Incluso dejando aparte la comodidad o autoconfianza del alumno, la intervención correctora del profesor es la de un agente externo al proceso mental del alumno, y para el aprendizaje activo sería preferible sacar el mayor partido posible de procesos internos.
- Puesto que el alumno sabe que el profesor conoce el algoritmo y le corregirá, inconscien-

temente puede descansar en este hecho asumiendo un papel de espectador; se caería en los defectos de la estrategia expositiva.

Por tanto, parece conveniente situar al alumno en una situación controlada, pero no por el profesor, para evitar los patrones habituales de pasividad.

Parece una buena opción recurrir a un programa de ordenador que ofrezca al alumno un entorno de simulación. Insistimos en que no se trata de un programa de simulación al uso en el que el alumno ve cómo funciona el algoritmo; lo que se pretende es comprometer al alumno en la ordenación. El programa no será capaz de ordenar un vector, sino que sus objetivos son:

- Representar visualmente el problema.
- Ofrecer posibilidades de interacción para que el alumno resuelva el problema.
- Facilitar que el alumno, de manera implícita, identifique las vías erróneas de solución.

El último punto merece una explicación especial. No se trata de que el programa avise al alumno cuando hace algo mal o le aconseje; estaríamos sustituyendo al profesor por la máquina, pero con el mismo patrón de corrección. Para promover al máximo el proceso de *descubrimiento*, hay que intentar que el propio alumno diagnostique sus errores. Así, no adoptará el papel semi-pasivo de avanzar esperando intervenciones, sino que podrá concentrarse en llegar a una solución, de la que él mismo es (casi) el único responsable, y conocerá los errores frecuentes.

Por supuesto, no cabe esperar que el alumno encuentre en todos los casos la solución por sí mismo, y (en caso estrictamente necesario) el profesor puede intervenir en su ayuda. Y lo que es más importante, aunque encuentre la solución y pueda ordenar el vector con las restricciones dadas, difícilmente podrá poner de relieve los diversos aspectos de su solución para implementar el algoritmo. Esta es la etapa en la que el profesor intervendrá para completar el proceso.

2.3. Condicionantes del entorno

La experiencia se lleva a cabo en un aula de teoría con un alto número de alumnos (normalmente entre 60 y 100). No hay ordenadores para uso individual, y además la intervención del profesor (mínima, pero no nula) no resultaría posible con tantos alumnos.

En lugar de poner a cada alumno a trabajar en solitario, se pide un voluntario cuyas acciones se muestran en una pantalla de vídeo, y los compañeros pueden seguirlas en todo momento. Esto presenta las siguientes ventajas:

- Se mantiene una cierta expectación (y por tanto atención) de la clase. Ver cómo un compañero busca la solución utilizando un programa (que tiene algo de *juego de ordenador*) resulta más llamativo que escuchar una exposición magistral del profesor. Además, los alumnos se identifican con las acciones del voluntario, y cuando este comete un error (que es parte importante del trabajo) se dan cuenta casi al unísono.
- El profesor mantiene un mayor control. Todos los alumnos están haciendo básicamente el mismo trabajo, y las referencias del profesor tienen sentido para todos los que hayan podido seguir el desarrollo de la clase.

Todas las clases de teoría de esta asignatura se imparten utilizando un ordenador y un cañón de vídeo (para mostrar código, diapositivas o elementos audiovisuales), por lo que los medios materiales están disponibles.

2.4. Desarrollo

Lógicamente, el primer requisito para llevar a la práctica esta estrategia es disponer del programa de ordenador que responda a los requisitos planteados.

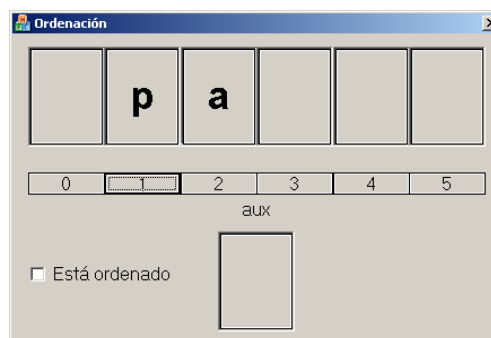


Figura 1. Simulador para algoritmos de ordenación

El programa desarrollado tiene el aspecto que se muestra en la Figura 1. No es dicho programa lo que interesa aquí, sino la estrategia didáctica a la que sirve de apoyo. Las características relevan-

tes para el aprendizaje por descubrimiento basado en errores son:

- En la parte superior puede verse un vector de caracteres, y bajo él las etiquetas de los índices. El vector es expresamente de caracteres para que los alumnos que encuentran dificultades en ello no confundan el contenido y los índices (enteros).
- En cada momento sólo es posible ver dos elementos (consecutivos) del vector. Los valores se descubren al pulsar con el ratón sobre el índice; en este caso, el alumno ha pulsado sobre el índice “1”.
- La casilla *aux* de la parte inferior es una variable auxiliar que el alumno puede utilizar. Para asignar un valor de una variable a otra, se hace clic con el ratón en la casilla de origen; en ese momento, el cursor del ratón cambia, tomando la forma de la letra que se ha “cogido”. Al hacer un segundo clic en la casilla de destino, esta toma el valor.
- La marca *Está ordenado* de la parte izquierda hace el papel de variable booleana (la bandera del algoritmo).

Al alumno se le explica el manejo básico del programa, y se le invita a ordenar el vector.

Por supuesto, cada alumno tendrá su propia estrategia, pero es frecuente que se incurra en lo siguiente:

- Al descubrir una pareja de valores y ver una a , toma el valor y acude rápidamente al elemento 0 para dejarla allí (o intenta llevar una z al final). Entonces se da cuenta de que no puede hacerlo directamente.
- Al intercambiar una pareja de valores, no pasa por *aux* y advierte que ha perdido el valor que había en la casilla de destino. También suele ocurrir que no realice el intercambio en el orden adecuado, perdiendo algún valor.
- Al llegar al final del vector, no está seguro de si está ordenado; debe volver a verificarlo.

A causa de los errores anteriores, es probable que el alumno deba empezar de nuevo algunas veces, pero en esos intentos irá adquiriendo como rutinarios movimientos que al principio no lo eran (y un proceso mental similar se produce en los espectadores). Cuando el alumno termina, puede sentarse, y el profesor llama la atención sobre algunos puntos de su actuación, resaltando lo *natural* de algunos errores y cómo la evitación de estos influye en la estructura del algoritmo:

- No tiene sentido hacer clic sobre el índice 5, puesto que no hay un elemento 6.
- El movimiento instintivo de elementos en saltos largos (por ejemplo, poner la a en la posición 0) no funciona en estas condiciones.
- Hay que ser muy cuidadosos con el intercambio de elementos, para no sobrescribir o perder un valor.
- Sabemos que el vector está ordenado cuando damos una pasada sin hacer ningún cambio.

Finalmente, se realiza la parte expositiva del código del algoritmo de manera constructiva, como una traducción de las acciones anteriores.

3. Ordenación por inserción directa

3.1. Fundamentos de la estrategia docente

Habitualmente, los alumnos encuentran aún más dificultades con el algoritmo de inserción directa, ya que incluye varios sub-algoritmos:

- Selección del elemento a insertar
- Búsqueda de su nueva ubicación
- Desplazamiento de los demás elementos para “hacer sitio”

Comprender cada uno de estos sub-algoritmos requiere un cierto esfuerzo. Por ejemplo, al desplazar un conjunto de elementos en un vector es frecuente que se empiece en la dirección errónea, de modo que el primer elemento desplazado se repita sobrescribiendo los demás. Esto sucede aunque a lo largo del curso se hayan hecho ya diversos ejercicios sobre el particular y se haya insistido en que para desplazar sub-vectores hay que “tirar” de los elementos y nunca “empujarlos”, es decir, iniciar el bucle desde la parte del vector a la que se dirige el desplazamiento.

Además de dificultades concretas como esta, es aún más difícil para el alumno asimilar completamente la relación que existe entre los componentes del algoritmo para solucionar el problema. Sin embargo, nuevamente el algoritmo es una solución con una clara faceta *natural* o intuitiva que sería interesante poner de manifiesto.

En este caso, la estrategia del error no parece apropiada. En la burbuja con señal comprende con relativa facilidad el método y falla al ponerlo en práctica; identificar los errores puede ayudarle. En la inserción directa el problema es ya de comprensión; se buscará que el alumno se aproxime al algoritmo apoyándose en su propia experiencia, para que por analogía identifique su estructura.

3.2. Diseño de la situación de aprendizaje

Nuevamente, se trata de ubicar al alumno en una situación en la que pueda llegar a la solución del algoritmo de manera natural. Se podría recurrir a otro programa de simulación, pero parece más indicada una actividad al margen de la informática, ya que tiene especial interés relacionar el algoritmo con la experiencia previa del alumno.

Frecuentemente se compara el algoritmo de inserción directa con las acciones de un jugador de cartas cuando las coloca en su mano; llevando esto a la práctica, se diseñó un ejercicio en el que efectivamente el alumno manipule cartas reales y pudiese ver de forma intuitiva el algoritmo en acción. A diferencia del algoritmo de burbuja, en el que se pretendía identificar errores y tenía especial importancia el proceso de corrección al alumno, para el algoritmo de inserción la importancia reside en que el alumno realice movimientos naturales y los identifique después con los elementos del algoritmo.

3.3. Condicionantes del entorno

Los condicionantes son los mismos que los descritos para el algoritmo de burbuja, y son aplicables las mismas consideraciones (con algunos medios adicionales que se describirán en el apartado siguiente). En este caso también se trabaja con un voluntario, cuya actuación siguen sus compañeros en una pantalla de vídeo.

3.4. Desarrollo

En primer lugar, y aun disponiendo de los medios mencionados anteriormente, es necesario instalar una cámara de vídeo en el ordenador que está conectado al cañón de vídeo. No son necesarias altas prestaciones, por lo que cualquier *webcam* puede servir. Es necesario también tener un programa que permita ver en tiempo real la imagen de la cámara y simultáneamente grabar el vídeo, ya que las precisiones finales del profesor son importantes y conviene hacerlas sobre la misma secuencia que los alumnos han visto, para conseguir un mayor refuerzo de los conceptos. En el ejercicio se utiliza también una baraja.

Una vez preparada la cámara, como en el caso anterior se pide un voluntario. Se le dan las instrucciones siguientes:

- El profesor le irá suministrando cartas (todas del mismo palo), una por una.
- El alumno debe ir colocándolas en fila (y en orden), y esta fila debe estar en todo momento “apoyada” en el borde izquierdo de la mesa.
- Durante toda la operación, el alumno sólo puede utilizar un dedo para desplazar las cartas por la mesa.

Entonces el profesor activa la cámara y muestra (registrándolo a la vez) lo que ocurre en la mesa. Coloca una carta en la mesa, que el alumno debe llevar a su lugar arrastrándola con el dedo. Es conveniente que las 3 ó 4 primeras cartas se suministren en orden; el alumno se limita a añadirlas por la derecha (Figura 2, capturada de una sesión real).

Entonces se suministra una carta que deba ir antes de algunas de las que ya están en la mesa. El alumno, para introducirla en su lugar, deberá desplazar las demás, y a fin de que no se solapen unas con otras, por lo general¹ empezará instintivamente el desplazamiento por la derecha, creando huecos (“tirando” de los elementos y no “empujándolos”), como se ve en la Figura 3.

Se puede continuar este proceso con algunas cartas más; hecho esto, el alumno puede volver a su sitio. El profesor detiene la grabación, y empieza la fase de recapitulación en la que ejerce su papel orientador llamando la atención sobre la relación entre ciertas actuaciones del alumno y las diversas partes del algoritmo:

- En primer lugar, se tiene la nueva carta (el valor a insertar) en una “variable auxiliar” (en otro lugar de la mesa).
- Aunque esta acción no se vea (es un proceso mental) el alumno elige el lugar en el que debe ir la nueva carta (lo que equivale a una búsqueda secuencial en estructura ordenada).
- Una vez elegida la posición, el alumno desplaza todas las cartas que hay a la derecha de dicha posición, para “crear un hueco”.
- Para desplazarlas, empieza a mover la carta de más a la derecha (bucle descendente).
- Una vez desplazadas, coloca la nueva carta (que estaba en la “variable auxiliar”) en el hueco creado.

¹ Por lo general. En caso contrario, no hay más remedio que intervenir para evitar la “trampa”.

- Si la nueva carta es mayor que todas, simplemente se coloca a la derecha (se desplazan 0 cartas).
- A cada nueva inserción, el vector ordenado va creciendo.

Al mencionar cada uno de estos elementos, el profesor recurre al vídeo que ha grabado, mostrando el fragmento en el que aparece. Hecho esto, relaciona estas ideas con el código del algoritmo.

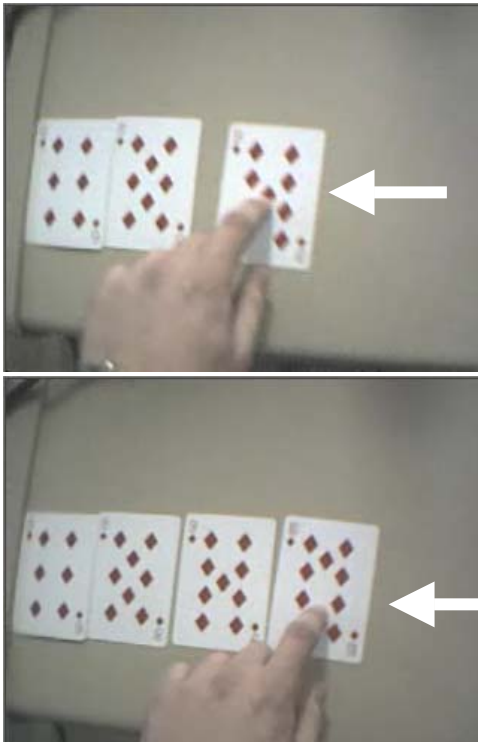


Figura 2. Añadiendo cartas por la derecha - secuencia ordenada (6, 8, 9 y 10 de diamantes)

En realidad, con todo este proceso se ha descrito sólo el mecanismo de inserción ordenada. Para completar la versión del algoritmo de ordenación interna, sólo falta hacer ver al alumno que las cartas que el profesor iba poniendo sobre la mesa de una en una podían estar ya sobre ella, en una secuencia desordenada; en ese caso, su compañero habría ido tomando una de cada vez y poniéndola en la variable auxiliar, pero el resto del proceso sería idéntico. Esta última parte resulta un tanto oscura para los alumnos.

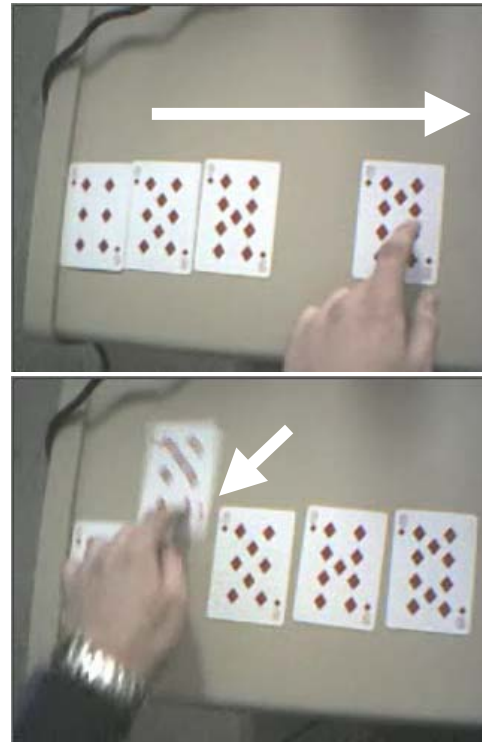


Figura 3. Desplazando cartas a la derecha e insertando la nueva carta (7 de diamantes)

4. Resultados

El propósito de esta experiencia era simplemente definir y poner en práctica unos métodos docentes alternativos y operativos para sustituir a la lección expositiva y promover un papel más activo del alumnado. Se ha verificado su viabilidad y efecto mediante la observación en clase [9]. A la vista de los objetivos planteados, parece claro que se consigue una actitud más activa del alumno, una mayor atención, un enfoque más intuitivo de esta materia y el establecimiento de una relación entre los algoritmos y experiencias propias del alumno.

El mayor problema en la práctica es, curiosamente, conseguir voluntarios, a pesar de que se insiste en que se *necesita* que el alumno no tenga conocimientos previos de la materia. La infraestructura, si se ha preparado, no da problemas, y tampoco ha habido un impacto temporal negativo.

Respecto a resultados cuantitativos, parece difícil detectar mejoras significativas en los indi-

ces de éxito o de asistencia para asignaturas (como la que nos ocupa) en las que tantos factores ajenos a las clases tienen una influencia decisiva. La utilidad de estas técnicas docentes depende, más que de una aplicación aislada que tendrá un efecto mínimo, de una línea de actuación constante. Creemos que son nada más (y nada menos) que un paso en la dirección correcta, un ingrediente más de una forma de entender la docencia, y en este marco sí que puede deducirse, tanto de las evaluaciones de los alumnos como de sus comentarios en diversos foros, que perciben de manera global una mejora en la calidad de la docencia y aumenta su grado de satisfacción, aun cuando los resultados de la evaluación sumativa puedan ser similares.

5. Conclusiones

En este artículo se han presentado sendos métodos didácticos para la enseñanza de dos algoritmos de ordenación, con las características siguientes:

- Se pretende experimentar con un método más atractivo y participativo que la lección expositiva.
- No hay exposición previa, sino que el alumno construye primero la solución de manera intuitiva, por descubrimiento, y luego esta se analiza.
- Para la ordenación por burbuja se aborda una estrategia de identificación de errores. Para la inserción directa, se realiza una síntesis constructiva.
- La lección tiene tres partes: ejercicio práctico (con un voluntario), análisis de los gestos y procesos implicados en el ejercicio, y reflejo de estas ideas en código (sólo al final).

Además de las ventajas que quepa encontrar en la consecución de los objetivos específicos aquí planteados, de manera indirecta este cambio en la docencia puede contribuir a mejorar la percepción general del alumno de las clases de teoría, al introducir una mayor variedad y participación.

Inciendo en esta línea, parece viable ampliar el uso de simuladores para su aplicación a los algoritmos de búsqueda (y así lo hemos puesto ya en práctica para el caso de la búsqueda secuencial y binaria), de modo que los alumnos aprecien también de forma natural la solución que aportan tales algoritmos.

Referencias

- [1] Charles C. Bonwell, James A. Eison. *Active Learning: Creating Excitement in the Classroom*. ASHE-ERIC Higher Education Report, No.1, Washington D.C.: The George Washington University, School of Education and Human Development, 1991.
- [2] Metodología de la Programación: sitio web. <http://www.euitio.uniovi.es/~mp>
- [3] Juan de Pablos Pons. *Estrategias de enseñanza y medios instructivos*. En *El trabajo en el aula*, Juan de Pablos Pons (Ed.), Ediciones Alfar, Sevilla, 1988. ISBN: 84-86256-60-7
- [4] Plan de Estudios 2002 – Escuela Universitaria de Ingeniería Técnica en Informática de Oviedo. <http://www.euitio.uniovi.es/nuevoplan>
- [5] Raquel Rodríguez, Jesús Hernández, Ana Alonso, Eliseo Díez-Itza. *El absentismo en la Universidad: resultados de una encuesta sobre motivos que señalan los estudiantes para no asistir a clase*. Aula Abierta - nº 82, Oviedo, Diciembre de 2003.
- [6] Raquel Rodríguez, Jesús Hernández, Samuel Fernández (coords.) *Docencia universitaria. Orientaciones para la formación del profesorado*. Universidad de Oviedo, 2004. ISBN: 84-88828-23-3
- [7] Ann Stallheim-Smith. *Focusing on Active, Meaningful Learning*. IDEA Paper nº. 34, IDEA Center, Kansas State University, 1998.
- [8] Saturnino de la Torre y Oscar Barrios (coords.) *Estrategias didácticas innovadoras. Recursos para la formación y el cambio*. Ediciones Octaedro, Barcelona, 2000. ISBN: 84-8063-412-X
- [9] D. B. Van Dalen, W. J. Meyer. *Manual de técnica de la investigación educacional*. Versión en español de *Understanding Educational Research. An introduction*. Editorial Paidós, 1981. ISBN: 84-7509-109-1