



Modelos de la teoría de grafos aplicados a problemas de competiciones de programación*

Cristina Jordán¹, Jon Ander Gómez², Marcos Calvo² y J. Alberto Conejero¹

¹Dpto. Matemática Aplicada. ETS Ingeniería Informática. Universitat Politècnica de València

²Dpto. Sistemas Informáticos y Computación. ETS Ingeniería Informática. Universitat Politècnica de València

Abstract

The subject of Algorithms for Problem Solving of the degree of Computer Science Bachelor's Degree of the ETSINF is geared towards the solution of programming challenges that are usually taken from programming competitions, such as the Southwestern Europe Regional Contest (SWERC), where students from the ETSINF have been regularly participating for the last years. The solution of such a problem is obtained by building a suitable mode for it, finding the optimal solution via this model, and being able of programming it without bugs in a short period of time. The skill on the solution of these problems is very much taken into account in recruiting processes of big technological companies such as Google, Apple, Yahoo, Microsoft or Facebook.

We show a collaboration between two subjects of this degree: Algorithms for Problem Solving (CP) and Graphs, models, and applications (GMA). This collaboration was proposed by students who had taken both subjects simultaneously. The goals consist on redirect part of the contents of GMA to the analysis of models that usually appear in this type of problems, and to facilitate that students can face this challenges. The methodology consists on raising several problems from the point of view of both subjects. The first impressions concerning the innovation are positive.

Keywords: *Algorithmics, graph theory; programming; programming challenges; SWERC.*

*Proyecto financiado por la Universitat Politècnica de València. PIME-B08

Resumen

La asignatura *Competición de Programación del Grado de Ingeniería Informática* está orientada a la resolución de desafíos de programación que se suelen proponer en competiciones como la *Southwestern Europe Regional Contest (SWERC)*, en la que alumnos de la *ETSINF* llevan participando asiduamente durante los últimos años. Para obtener la solución de un problema de este tipo se necesita hacer una modelización adecuada del mismo, así como hallar una solución óptima por medio del modelo y ser capaz de programarla sin errores en un corto espacio de tiempo. La habilidad en la resolución de dichos problemas se tiene muy en cuenta en los procesos de selección de personal de grandes compañías tecnológicas como *Google, Apple, Yahoo, Microsoft* o *Facebook*.

Mostramos una colaboración entre dos optativas de este grado: *Competición de Programación (CP)* y *Grafos, Modelos y Aplicaciones (GMA)*. Esta colaboración fue propuesta por alumnos que cursaban ambas asignaturas simultáneamente. Los objetivos consisten en reorientar parte de los contenidos de *GMA* al análisis de modelos que suelen aparecer con frecuencia en problemas de competiciones de programación, facilitando así los estudiantes para estos afrontar estos desafíos. La metodología consiste en plantear varios problemas desde la óptica de ambas asignaturas. Las primeras valoraciones de la innovación son positivas.

Keywords: *Algorítmica; teoría de grafos; programación; retos de programación; SWERC.*

1 Introducción-Contexto

Recientemente, uno de los criterios diferenciadores más utilizados por empresas del sector de las Tecnologías de la Información y las Comunicaciones, más allá de la nota media del expediente, es la participación en competiciones de programación. Es evidente que los logros que algunos alumnos alcanzan en las mismas son, en general, un claro ejemplo del nivel de sus habilidades en relación al de sus compañeros de titulación, pero con carácter general suponen un buen entrenamiento para la preparación de las entrevistas relacionadas con programación. Para el lector interesado en información general sobre los procesos de selección en grandes compañías como *Google, Yahoo, Apple* o *Microsoft*, referimos a *Laakman McDowell (2010)*, véase también *Bock (2015)*. Información más detallada sobre la preparación de la parte de programación en estas entrevistas puede ser encontrada en *Aziz, Lee y Prakash (2015)* y en *Laakman McDowell (2015)*. Por último, remitimos también a los excelentes libros de *Poundstone* sobre cuestiones no convencionales que se plantean como desafíos a los candidatos en las entrevistas de procesos de selección *Poundstone 2012* y también *Bock (2015)*.

La asignatura de *Competición de Programación* (CP) en la ETSINF responde a la necesidad que tienen los egresados de dominar la programación para superar los procesos de selección de personal en empresas de software. Detrás de las principales competiciones internacionales de programación están empresas como Google, Apple, Yahoo, Microsoft o Facebook. Y son muchas las empresas de ámbito internacional, además de las citadas, que seleccionan a sus ingenieros de software de entre los que participan en las competiciones de programación. El temario de esta asignatura cubre los aspectos más relevantes de la Algorítmica, aspectos que el alumno deberá utilizar y combinar adecuadamente para resolver los problemas planteados.

La asignatura de *Grafos, Modelos y Aplicaciones* (GMA) está orientada a la modelización de problemas que pueden ser resueltos mediante algoritmos, algunos de los cuales se ve su implementación en CP. A instancias de alumnos que cursaban ambas asignaturas y preparaban competiciones de programación, los profesores de ambas asignaturas decidieron colaborar para plantear algunos problemas desde ópticas distintas, el modelizado matemático y la algorítmica.

La presente comunicación refleja estos trabajos de colaboración y cómo se han planteado, indicando algunos ejemplos de las contribuciones de cada asignatura a la resolución de ciertos problemas. En la Sección 2 se revisa la estructura de la competición SWERC, que es la que se tiene en la ETSINF como referencia. En la Sección 3 se enmarcan ambas asignaturas en el vigente plan de estudios del Grado en Ingeniería Informática ofertado por la ETSINF. La innovación se plantea en la Sección 4. Algunos ejemplos de problemas considerados por ambas asignaturas, desde sus respectivas ópticas se hallan en la Sección 5. Finalmente, las conclusiones y futuras líneas de trabajo conjunto están contenidas en la Sección 6.

Colaboraciones entre asignaturas tecnológicas y básicas pueden darse en otras titulaciones en el marco de participación en otras competiciones o certámenes dirigidos a alumnos de grado y máster. De esta manera, los alumnos trabajan en aplicaciones auténticas de ciertos contenidos teóricos, valorando la utilidad de la abstracción y de la modelización. El interés de participación en las mismas es una forma óptima de enlazar competencias específicas, ligadas a la propia titulación, con competencias transversales, como pueden ser liderazgo, trabajo en equipo, o comunicación efectiva.

2 La SWERC

Las competiciones de programación suelen ser organizadas por las propias escuelas y facultades de informática o por grandes compañías como Google o Facebook. Desde hace años, la Escuela Técnica Superior de Ingeniería Informática de la Universitat Politècnica de València (ETSINF) viene participando de manera regular en la *International Collegiate Programming Contest*, que es la más antigua de estas competiciones y que está promovida por la Association for Computer Machinery (ACM) y patrocinada por IBM. Esta competición tiene varias fases. Cada escuela participante suele realizar una competición interna para seleccionar los miembros de los equipos que la representen, suelen ser entre 1 y 3. Estos equipos participan entonces en una fase

regional. En particular, las universidades españolas participan en la Southwestern Europe Regional Contest (SWERC), junto con universidades de Italia, Portugal, Suiza, Francia y Austria. A la final mundial acceden los ganadores de cada una de las más de 90 competiciones regionales que tienen lugar por todo el mundo.

El esquema de competición en cada uno de los niveles es bastante similar. En ella compiten varios equipos o personas (si es individual). A cada uno de ellos se le entrega una serie de problemas. Gana el que resuelve correctamente más, y en caso de empate, el que tarda menos tiempo, teniendo en cuenta las penalizaciones que se asignan a las soluciones incorrectas a los problemas para los que al final se halla una solución correcta. Uno de los textos que se suele considerar como base para preparar la participación en esta competición es el manual de Skiena y Revilla 2003.

Los participantes disponen de un ordenador en el que se ha instalado y configurado un sistema operativo Linux con una configuración básica para la programación en Java y C++. Estos ordenadores sólo están conectados con un ordenador que hace las veces de juez. Cuando un participante considera que ha resuelto un problema lo envía al juez, que compila la solución propuesta, comprueba si el resultado que se da es el correcto y además si la solución ha sido programada de manera eficiente (dentro de un tiempo y recursos determinados). Los participantes disponen de 5 horas para resolver los desafíos. Cada envío incorrecto penaliza con 20 minutos, que sólo se computan si el problema finalmente es resuelto. Detalles sobre la estructura y organización de este evento pueden encontrarse en Gómez 2013, en el que se detalla la organización de la SWERC de los años 2012 y de 2013 en la ETSINF.

3 Plan de estudios y las optativas

La solución de estos problemas está basada principalmente en el dominio de los contenidos de las asignaturas de programación y algorítmica. En el vigente plan de estudios del Grado en Ingeniería Informática impartido por la ETSINF se recogen en las siguientes asignaturas.

- *Introducción a la informática y la programación* (6 ECTS - 1er curso)
- *Programación* (6 ECTS - 1er curso)
- *Estructuras de datos y algoritmos* (4.5 ECTS - 1er curso)
- *Algorítmica* (4.5 ECTS - 3er curso, Intensificación de computación)
- *Competición de Programación* (4.5 ECTS - 4o curso, optativa)

Con respecto a los anteriores planes de estudios de Ingenierías Técnicas y Superiores en Informática, en el grado vigente se ha constatado una reducción de los créditos destinados a estos contenidos. La asignatura de Competición de Programación sirve tanto de revisión y profundización de conceptos ya vistos anteriores, como para la presentación a los alumnos de nuevos algoritmos más específicos. La metodología de

evaluación de esta asignatura está basada en la resolución de problemas de competiciones de programación de entre una lista propuesta a principio de cada año. Los problemas están ordenados por dificultad. Las correcciones de las soluciones de los alumnos, al igual que en las competiciones, se hacen enviando las soluciones a un juez.

Adicionalmente, el dominio de ciertos contenidos matemáticos es conveniente a la hora de poder plantear con éxito la resolución de un problema. En ese sentido, las asignaturas más estrechamente vinculadas son:

- *Matemática Discreta*, en particular las prácticas que están dedicada al estudio de los principios básicos de la teoría de grafos (1.5 ECTS - 1er curso)
- *Estadística* (6 ECTS - 1er curso)
- *Grafos, Modelos y Aplicaciones* (4.5 ECTS - 4o curso, optativa)

Cabe destacar, que pese al interés creciente de las empresas en las habilidades de los egresados en estos contenidos, su presencia en las asignaturas obligatorias se ha visto reducida en créditos en el actual plan de estudios, con respecto a los anteriores planes de estudios de Ingeniero Técnico en Informática (especialidades de Gestión y de Sistemas y Redes) y al primer ciclo de Ingeniero en Informática.

4 Innovación

4.1 Punto de inicio

La innovación propuesta surge respondiendo a una petición de alumnos que se encontraban cursando las optativas de CP y GMA, que percibieron que el formalismo en la modelización de los problemas en GMA les facilitaba plantear la resolución de algunos problemas de CP.

Por una parte, el objetivo fundamental de la asignatura de CP es *aprender a programar de forma eficiente y con tiempo limitado*, a partir de la resolución de problemas de muy distinta índole y dificultad, siendo una buena parte de estos resolubles desde la teoría de grafos. Por otra parte, en GMA se plantea como uno de sus objetivos *la resolución algorítmica de problemas relativos a la vida real, tipificados, previa transformación en problemas teóricos en el ámbito de la teoría de grafos*. Esta transformación del problema es lo que se conoce como *modelización matemática*. Por todo ello, aunque las asignaturas GMA y CP tengan perfiles muy diferentes, el enfoque dado a GMA resulta muy útil como apoyo a la resolución de problemas de CP.

4.2 Planteamiento y desarrollo

La innovación planteada se enmarca en el desarrollo normal de ambas asignaturas que se imparten simultáneamente en el segundo semestre del cuarto curso del Grado en Ingeniería Informática. Se ha planteado de manera que los alumnos que cursan al menos una de las dos asignaturas puedan beneficiarse, aunque sobre todo los que estén cursando las dos. En esta primera experiencia nos encontramos con 30 alumnos matriculados en CCP y 14 en GMA, siendo 6 el número de alumnos comunes a ambas asignaturas.

En primer lugar se realizarán en cada asignatura sesiones abiertas en las que los profesores de cada asignatura aportan su contribución al desarrollo de ciertos problemas: desde GMA se estudia la modelización del mismo y desde CP se revisa la implementación de los algoritmos que retornan la solución a cada uno de los problemas planteados. Además, las clases de GMA, más teóricas, estarán soportadas mediante la grabación de las clases en la herramienta Videoapuntes de Poliformat, por lo que si algún alumno no puede acudir a sus sesiones abiertas, puede revisarlo después. Esto no supone una carga adicional, sino una oportunidad de tener el material disponible el material durante todo el período lectivo, así como durante el período de preparación de las competiciones de programación, tanto la local como la internacional.

Los alumnos de CP son evaluados mediante la resolución de problemas de entre una colección de 20 problemas propuestos, cuya solución se ha de enviar a un juez árbitro para su valoración. La evaluación de los alumnos de GMA consiste en entregas periódicas de cuestionarios relativos a los contenidos vistos en clase.

Se han seleccionado algunos problemas de CP y algunos cuestionarios de GMA, de manera que la resolución de los primeros o la entrega de los segundos contribuye simultáneamente a la nota de ambas asignaturas. En todos los casos, se eligen contenidos complementarios, algorítmica y modelización de un mismo problema.

Un alumno que sólo cursa una de ellas tiene así la oportunidad de cambiar el tipo de tarea que le contribuye a la evaluación de la asignatura que está cursando, es decir, si cursa CP puede entregar algún ejercicio de desarrollo más conceptual propio de GMA, y si sólo cursa GMA puede entregar un ejercicio más de programación ofrecido desde CP. Si un alumno cursa las dos, una misma tarea les sirve para contribuir a la nota de ambas asignaturas. Siendo generosos en este planteamiento, nuestra percepción preliminar es que una vez puestos en un problema, los alumnos prefieren continuar realizando tareas en torno al mismo, aunque tengan un enfoque distinto. La evaluación de contenidos que puede estar sujeta a esta iniciativa puede llegar a alcanzar el 20 por ciento de los contenidos de ambas asignaturas. En todos los casos, los alumnos se ven beneficiados, puesto que siempre pueden seguir cada asignatura de manera independiente a la otra, teniendo además la oportunidad de poder contar en la nota de una asignatura con la puntuación correspondiente a ejercicios adicionales de la otra.

5 Problemas ejemplo

Los problemas utilizados en cualquier asignatura son en unos casos elaborados por los profesores de la asignatura y en otros extraídos de libros, apuntes, web, etc. En nuestro caso, dada la relación entre ambas asignaturas, aparte de problemas propios, planteamos problemas de la base de datos utilizada por la asignatura de programación *ACM Contest Problems Archive*, <https://uva.onlinejudge.org/> de la Universidad de Valladolid, así como el libro de Skiena y Revilla [2003](#).

Antes de presentar una muestra de estos problemas, en aras de la claridad, vamos a hacer unos breves comentarios sobre la base de la modelización utilizando grafos. Un grafo es una pareja de conjuntos V (siempre distinto del conjunto vacío) y E , conjunto de vértices y aristas respectivamente, donde una arista es un par de elementos de V . Gráficamente, los vértices se representan con puntos y las aristas con líneas. Más concretamente si el par (u, v) es distinto del par (v, u) diremos que el grafo es dirigido y su representación gráfica es una flecha de u hacia v , mientras que si son iguales lo representaremos mediante una línea entre u y v . Para más información sobre teoría de grafos y los algoritmos a los que haremos mención a continuación referimos, entre otros, a Christofides ([1975](#)), Gross y Yellen ([2004](#)), Jordán y Torregrosa ([1996](#)) y Conejero y Jordán ([2015a](#)); Conejero y Jordán ([2015b](#)).

Cuando nos enfrentamos a un problema real debemos buscar qué concepto de los utilizados se va a representar como vértice y qué concepto como arista. Una vez modelizado el contexto del problema como un grafo transformamos cada uno de los datos, circunstancias, condiciones, etc. del problema en conceptos de la teoría de grafos, de manera que una vez convertido el problema en un problema abstracto podamos resolverlo utilizando la teoría estudiada. Tanto la elección de qué se va a representar como vértices y qué como aristas y qué es lo que se desea encontrar o resolver en términos de grafos están íntimamente ligados. Es el proceso que conocemos como modelización y puede ser muy sencillo o muy complicado ya que el resultado de la modelización puede ser un problema estándar ya estudiado (con lo que será suficiente aplicar la solución conocida), semejante a un problema estándar (por lo que habrá que modificar un método conocido para que se adecúe a nuestras circunstancias) o, sencillamente, un problema nuevo para el que tendremos que aportar una solución imaginativa. Obviamente, cuanto mayores sean nuestros conocimientos en modelización más fácil nos resultará dar con una solución apropiada. Presentamos a continuación tres ejemplos y la modelización y algoritmos a aplicar en cada caso.

Problema del collar roto (Skiena y Revilla [2003](#))

Mi hermana pequeña tiene un bonito collar hecho de abalorios, cada uno de ellos consta de dos partes, cada una de un color. Dos abalorios sucesivos cualesquiera comparten un color común en su punto de contacto. Pero un día el collar se rompió y todos los abalorios quedaron esparcidos por el suelo. Mi hermana hizo lo que pudo para recogerlos pero no está segura de haberlos recogido todos.

Ahora es cuando entramos nosotros. Ella quiere saber si es posible hacer un nuevo collar utilizando todos los abalorios que tiene, siguiendo el modelo del collar original. En caso de que sea posible, ¿cómo habría que colocar los abalorios?

Solución: En este caso consideraremos como vértices los colores y como aristas las cuentas. Como un collar es un encadenamiento de cuentas, el problema se traduce en determinar si existe lo que se conoce como una cadena en el grafo definido, con la condición de que ninguna de las aristas de la cadena se repita (dado que cada cuenta solo puede aparecer una vez en el collar).

Este tipo de cadenas recibe el nombre de cadenas eulerianas no cerradas. Su existencia queda determinada por una conocida caracterización. En caso de existir encontraremos la solución mediante la aplicación de una modificación del algoritmo de Hierholzer.

Paseando con Ralph (ACM Contest Problems Archive, <https://uva.onlinejudge.org/>).

Bob pasea frecuentemente con su perro Ralph. Bob anda a velocidad constante, siguiendo una línea poligonal cuyos vértices P_i vienen especificados por 8 puntos de coordenadas cartesianas (x_i, y_i) , $i = 1, 2, \dots, 8$, que son $P_1 = (0, 0)$, $P_2 = (2, 0)$, $P_3 = (2, 2)$, $P_4 = (1, 1)$, $P_5 = (3, 1)$, $P_6 = (3, 3)$, $P_7 = (1, 3)$, $P_8 = (0, 2)$.

Ralph pasea por su cuenta, pero encontrándose siempre con su dueño en los n puntos antes comentados. Ambos empiezan a caminar simultáneamente en el punto $P_1 = (x_1, y_1)$ y terminan en dicho punto.

El ritmo de Ralph es hasta dos veces el de Bob. Mientras este camina en línea recta siguiendo el segmento que une los puntos $P_i = (x_i, y_i)$ y $P_{i+1} = (x_{i+1}, y_{i+1})$, el perro husmea entre los árboles, busca conejos, ... en sitios "interesantes" de las zonas cercanas que él conoce. Estas vienen marcadas como 11 puntos Q_i de coordenadas (x_i, y_i) , $i = 1, 2, \dots, 9$. (Ver puntos Q_i al final del problema).

Sin embargo, después de dejar a Bob en el punto $P_i = (x_i, y_i)$ y antes de encontrarse de nuevo con él en $P_{i+1} = (x_{i+1}, y_{i+1})$, Ralph visita como máximo uno de sus m sitios Q_i preferidos. Ralph nunca llega al punto de encuentro más tarde que Bob.

Los puntos son $Q_1 = (3, 0)$, $Q_2 = (1, 2)$, $Q_3 = (4, 2)$, $Q_4 = (-1, 1)$, $Q_5 = (2, 5, 2, 5)$, $Q_6 = (1, 5, 0, 5)$, $Q_7 = (0, 5, 1, 5)$, $Q_8 = (0, 5, 0, 5)$, $Q_9 = (1, 4)$.

Encuentra la ruta de Ralph que satisfaga las condiciones anteriores y le permita visitar el mayor número posible de sitios interesantes”.

NOTA: Recordad que la distancia entre dos puntos $x = (x_1, x_2)$ e $y = (y_1, y_2)$ viene dada por la fórmula $d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$

Solución: La modelización consiste en considerar como vértices los puntos P_i y Q_i del enunciado. En cuanto al conjunto E , supondremos que existe una arista entre un vértice Q_j y un vértice P_i si $d(P_i, Q_j) + d(Q_j, P_{i+1}) \leq 2d(P_i, P_{i+1})$.

Tal como se han definido las aristas y atendiendo al enunciado el objetivos es emparejar los sitios interesantes para Ralph, Q_i , con el punto inicial, P_i , de los segmentos de la línea poligonal que describe Bob en su paseo, o lo que es lo mismo, con cada uno de dichos segmentos.

En términos de la teoría de grafos, nos piden que encontremos un emparejamiento máximo, para lo que aplicaremos el algoritmo de Edmonds.

Robots centinela (*Competición local de programación. ETS Ingeniería Informática UPV. Problema fue propuesto por Ximo Planelles.*).

Necesitamos vigilar una serie de puntos interesantes usando robots centinela que no pueden moverse ni girar. Podemos situar un centinela en cualquier posición, ya sea norte, sur, este u oeste. Una vez fijado el robot en su posición, controlará los puntos de interés que se encuentren frente a él.

Para vigilar dos o más puntos ubicados en la misma fila o columna basta con un único centinela. Desgraciadamente, hay también obstáculos a través de los cuales el robot no puede ver.

Dados un conjunto de puntos de interés y obstáculos en una rejilla, se pide calcular el mínimo número de robots centinela necesarios para vigilar todos los puntos. Téngase en cuenta que para que un robot pueda controlar un punto de interés este debe encontrarse enfrente y no haber ningún obstáculo entre ellos.

Solución: En este problema consideraremos dos tipos de vértices. Por un lado, el conjunto X formado por los posibles robots y por otro el conjunto Y de los sitios interesantes.

El conjunto de vértices V es la unión de los conjuntos anteriores. Supondremos que (u, v) es una arista de G si u pertenece a X , v pertenece a Y y además el robot u puede vigilar el punto v . En este caso, tras eliminar los posibles vértices en los que no incide ninguna arista, consideraremos que el grafo es dirigido y generaremos una red asociada en la que las aristas entre el vértice fuente de la red y los vértices de X tengan asociado un peso infinito, mientras que las aristas entre los vértices de Y y el sumidero por una parte y los arcos de vértices de X a vértices de Y tengan asociado un uno cada uno de ellos. El algoritmo de Ford-Fulkerson, que proporciona un flujo máximo, nos dará la solución.

6 Conclusiones

En esta comunicación hemos visto la utilidad de la colaboración de una asignatura eminentemente práctica y otra más teórica en aras de mejorar la preparación de los alumnos para el mercado laboral. Cabe precisar que si bien los contenidos de ambas asignaturas no son de una aplicación directa y diaria en el desarrollo de muchas actividades laborales, están resultando críticos a la hora de acceder al mercado laboral, sobre todo en compañías tecnológicas de reconocido prestigio. En este sentido, cabe recordar que uno de los objetivos principales de cualquier grado universitario es formar a los futuros egresados para que puedan seguir aprendiendo por ellos mismos a lo largo de la vida, y en buena medida esta capacidad es la buscada por las compañías cuando valoran los resultados en competiciones de programación.

La innovación presentada continúa desarrollándose durante el segundo cuatrimestre del curso 2015-2016, buscando y ampliando la cantidad de problemas en los que ambas asignaturas pueden colaborar. Asimismo, estamos pendientes de hacer una valoración de los resultados de la innovación mediante el pase de encuestas a los alumnos de ambas asignaturas, tanto a los que han seguido esta innovación como a los que no, con el fin de valorar adecuadamente su impacto. Nuestras primeras impresiones son que la valoración de los alumnos involucrados es muy positiva y esperamos que dicho estudio confirme estas primeras sensaciones, en cualquier caso, la recopilación de esta información nos permitirá detectar los puntos mejorables de nuestra actuación. Esperamos que en cursos posteriores aumente el número de estudiantes que se matricula en ambas asignaturas.

Por último, queremos resaltar que esta colaboración no se limita al contexto de ambas asignaturas, sino también a la posterior preparación de los alumnos de los equipos preseleccionados que participan anualmente en la SWERC.

Entendemos que esta experiencia es un buen ejemplo de cómo la colaboración entre profesores de distintos departamentos puede enriquecer notablemente la calidad de la enseñanza de las asignaturas implicadas al mismo tiempo que se mejora la formación de los alumnos y disminuye la dispersión de contenidos.

Agradecimientos

Los autores quieren agradecer a la dirección del centro su compromiso con la iniciativa y con el patrocinio de la participación de equipos de alumnos de la escuela en la SWERC durante los últimos años, así como a los alumnos de ambas asignaturas que nos sugirieron esta colaboración.

Referencias bibliográficas

- Aziz, Adnan, Tsung-Hien Lee y Amit Prakash (2015). *Elements of programming interviews: The insider's guide*.
- Bock, Laszlo (2015). *Work rules!* Hachette Book Group.
- Christofides, Nicos (1975). *Graph Theory: An Algorithmic Approach (Computer Science and Applied Mathematics)*. Orlando, FL, USA: Academic Press, Inc. ISBN: 0121743500.
- Conejero, J. Alberto y Cristina Jordán (2015a). *Aplicaciones de la Teoría de Grafos a la vida Real I*.
- (2015b). *Aplicaciones de la Teoría de Grafos a la vida Real II*.
- Gómez, Jon A. et al (2013). “Competiciones de programación. Estímulo y salida laboral para los alumnos”. En: *Actas de las XIX Jenui*. Ed. por Universitat Jaume I, págs. 161-166.
- Gross, Jonathan L. y Jay Yellen (2004). *Handbook of graph theory*. Discrete mathematics and its applications. Boca Raton (Fla.), London, New York: CRC press. ISBN: 1-58488-090-2.
- Jordán, Cristina y Juan Ramón Torregrosa (1996). *Introducción a la Teoría de Grafos y sus algoritmos*. Universitat Politècnica de València.
- Laakman McDowell, Gayle (2010). *The Google Résumé: How to prepare for a career and land a job at Apple, Google or any top tech company*. John Wiley & Sons, Inc.
- (2015). *Cracking the coding interview. 189 programming questions & solutions*. 6th. CareerCup.
- Poundstone, William (2012). *Are you smart enough to work at Google? Fiendish puzzles and impossible interview questions from the world's top companies*. Oneworld Publications.

Skiena, Steven S. y Miguel Revilla (2003). *Programming Challenges: The Programming Contest Training Manual*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 0387001638.

Valladolid, Univ. de. *ACM Contest Problems Archive*, <https://uva.onlinejudge.org/>.