

Actas de las XIX Jenui. Castellón, 10-12 de julio 2013  
ISBN: 978-84-695-8051-6 DOI: 10.6035/e-TiIT.2013.13  
Páginas: 161-166

## Competiciones de programación. Estímulo y salida laboral para los alumnos

Jon A. Gómez, Joaquín Planells, Assumpció Casanova, Mabel Galiano, Marisa Llorens, Germán Moltó, Francisco Marqués, Natividad Prieto, Francisco Álvaro, Antonio Barella, María José Castro-Bleda, Mercedes García, Álvaro Hermida, Carlos Herrero, Carlos D. Martínez-Hinarejos, Antonio Molina, Lidia Moreno, Joan Pastor, Moisés Pastor, Mario Rodríguez, Andrés Terrasa, Emilio Vivancos

Departament de Sistemes Informàtics i Computació  
Universitat Politècnica de València – 46022 València

{jon, xplanells, casanova, mgaliano, mllorens}@dsic.upv.es  
{gmolto, pmarques, nprieto, falvaro, tbarella}@dsic.upv.es  
{mcastro, mgarcial, ahermida, cherrero}@dsic.upv.es  
{cmartine, amolina, lmoreno, jpastor}@dsic.upv.es  
{mpastorg, mrodriguez, aterrassa, vivancos}@dsic.upv.es

### Resumen

Las competiciones de programación son un estímulo importante para los alumnos de informática. Durante los últimos años ha crecido el número de estas competiciones en su mayoría promovidas por grandes empresas con el propósito de reclutar alumnos brillantes.

La *International Collegiate Programming Contest* es la más antigua de ellas, promovida por la ACM y patrocinada por IBM. A la final mundial van los mejores equipos de cada una de las más de 90 competiciones regionales que tienen lugar por todo el mundo. La regional en la que pueden participar las universidades españolas es la *Southwestern Europe Regional Contest* (España, Italia, Portugal, Suiza, Francia y Austria). Esta regional fue organizada por la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València en su edición de 2012 y lo será también en 2013.

En este trabajo se presenta la experiencia de planificación de una regional, tanto a nivel organizativo como de planteamiento de los problemas para la competición. Asimismo, se destaca la importancia de este tipo de competiciones en cuanto al desarrollo de habilidades y destrezas en los alumnos, pues su dedicación a solucionar distintos tipos de problemas, combinando conocimientos de matemáticas con técnicas algorítmicas, les da una formación adicional que saben apreciar las grandes empresas. Algunos alumnos de la ETSInf que han participado en estas competiciones están en proceso o ya han sido contratados por empresas del sector como es Google.

### Abstract

Programming contests are a good stimulus for computer engineering students. These contests have experienced an expansion during last years thanks to the interest of big companies for hiring brilliant students.

The International Collegiate Programming Contest is the oldest one, promoted by ACM and sponsored by IBM. The ACM-ICPC World Final takes place every year, only the best teams can participate, which are selected from the ones who compete in the more than 90 regional contests celebrated around the world. Spanish universities can participate in the Southwestern Europe Regional Contest (Spain, Italy, Portugal, France and Austria). The 2012 edition of this regional contest took place in Valencia, organised by the School of Computer Engineering at the *Universitat Politècnica de València*. The 2013 edition will be organised in Valencia as well.

In this paper it is presented the experience of planning a regional contest, both from the point of view of organisation as from posing the problems. Additionally, it is pointed out the value of this sort of programming contests as they foster the development of some abilities and skills in students, because the time they spend solving problems where the knowledge of mathematical concepts and of algorithmic techniques is very important. This fact gives students a special training very appreciated by big software companies. Some former students from ETSInf who participated in international programming contests were hired or are in process of being hired by Google.

## Palabras clave

Programación, algorítmica, solución de problemas, salidas laborales.

## 1. Introducción

Las competiciones de programación van apareciendo progresivamente en el horizonte de los estudiantes de informática, con especial fuerza durante los últimos años, gracias al interés que han despertado en las grandes empresas de software. Éstas patrocinan la mayoría de las competiciones más conocidas internacionalmente. En algunas de estas competiciones la participación es únicamente presencial, como es el caso de la descrita aquí; en otras es remota durante las fases previas clasificatorias y presencial en la última fase.

El interés de las grandes empresas está bien definido: reclutar a los alumnos que destacan en este tipo de competiciones. Tal es el caso de exalumnos de la *Escola Tècnica Superior d'Enginyeria Informàtica* (ETSInf) de la *Universitat Politècnica de València* (UPV), que han sido contratados o están en proceso.

La más antigua de estas competiciones, con 37 ediciones, es la *ACM International Collegiate Programming Contest* (ACM-ICPC) [1]. Cada año se celebra la final mundial, en la que participan los equipos ganadores de cada una de las competiciones regionales que constituyen la fase clasificatoria. Para Europa existen actualmente cinco regionales, la *Southwestern Europe Regional Contest* (SWERC) [2] es la competición regional en la que pueden participar las universidades españolas junto con las de Italia, Portugal, Francia, Suiza y Austria.

Organizar este tipo de competiciones y preparar a los alumnos para su participación en ellas tiene como finalidad desarrollar en los alumnos involucrados destrezas y habilidades para afrontar problemas complejos de diversa índole, cuya solución requiere la combinación de diferentes fuentes de conocimiento, en especial de técnicas algorítmicas, y siempre con la eficiencia en el consumo de recursos como premisa de principio a fin. Actualmente, las grandes empresas se disputan a los alumnos que destacan en estas competiciones y el hecho de participar en ellas, aun sin destacar, es visto como algo muy positivo en los procesos de selección. Podríamos citar a muchas, pero son IBM, Google, Microsoft, Facebook, SAP y T-Systems, entre otras multinacionales, las que ofrecen puestos de trabajo bien remunerados para los alumnos con mayor destreza en programación, así como estancias durante los meses de verano.

Como profesores consideramos que focalizar el esfuerzo docente en mejorar el rendimiento del alumno

medio no debe estar reñido con facilitar el acceso a estas oportunidades a los alumnos destacados.

El resto del presente trabajo cubre en diferentes secciones los siguientes aspectos a destacar: una descripción de cómo se desarrolla una competición, la organización, la configuración del entorno de trabajo y del juez automático, la preparación del conjunto de problemas, el reparto de responsabilidades entre el personal involucrado, cómo los alumnos pueden entrenarse y, finalmente, las conclusiones.

## 2. ¿En qué consiste una competición de programación?

En una competición de programación en la que compiten varios equipos, o varias personas si es individual, el objetivo es resolver el mayor número de problemas dentro del tiempo establecido. En caso de empate a igual número de problemas se dirime por el tiempo utilizado. Cada vez que un problema es resuelto (aceptado por el juez automático) se acumulan los minutos utilizados en resolverlo, contando desde el inicio de la competición. Además, cada envío fallido penaliza unos minutos, 20 para el caso de la ACM-ICPC y sus regionales. Los minutos acumulados por envíos fallidos a cada problema sólo se computan cuando el problema ha sido resuelto, es decir, los minutos de penalización por envíos fallidos a un problema que un equipo no ha podido resolver no se acumulan.

<i>accepted</i>	AC
<i>compilation error</i>	CE
<i>memory limit exceeded</i>	MLE
<i>presentation error</i>	PE
<i>runtime error</i>	RE
<i>time limit exceeded</i>	TLE
<i>wrong answer</i>	WA

Cuadro 1: Veredictos que el juez automático emite tras evaluar cada envío.

Los motivos por los cuales un envío es rechazado son varios. El Cuadro 1 muestra los posibles veredictos. Es importante destacar que el veredicto TLE no implica que el código fuente enviado como solución al problema esté mal, sino que utiliza más tiempo de CPU del permitido para solucionar el problema en cuestión. El significado de MLE es análogo. Ambos límites son especialmente relevantes, pues no todas las soluciones correctas a un problema son aceptadas, únicamente las más eficientes. Este hecho convierte ciertos problemas aparentemente fáciles en auténticos retos. El código de error RE se da cuando el programa enviado falla durante su ejecución y PE se da cuando la solución está bien pero los datos no se presentan según el formato solicitado en el enunciado del problema.



Figura 1: Explicación de las soluciones a los problemas en SWERC 2012.

## 2.1. Desarrollo del evento

Una competición internacional como la SWERC se planifica para un fin de semana completo. Sábado: registro, charlas científico-técnicas de interés para los alumnos, sesión de prueba durante dos horas y cena. Domingo: competición, explicación en líneas generales de cómo se podía resolver cada problema y entrega de premios.

Durante la sesión de prueba, que sirve a los participantes para tomar contacto con el sistema, deben resolverse tres problemas muy sencillos. A los administradores les permite comprobar que todas las configuraciones funcionan correctamente y a los jueces humanos cerciorarse de que el juez automático funciona sin problemas.

La competición dura cinco horas y los participantes deben resolver hasta diez problemas. Durante las cinco horas el equipo de jueces humanos está controlando el desarrollo de la competición y comprobando los envíos de cada equipo participante. Sobre todo se revisan los envíos con el veredicto WA a efectos de comprobar que una buena solución no es descartada por un fallo en la preparación de los casos de prueba. Ésta es la mayor preocupación del equipo de jueces humanos desde que empieza la competición hasta tener validada la clasificación final. También es responsabilidad de los jueces humanos atender las reclamaciones que puedan haber presentado algunos equipos. Y todo ello con apremio,

dado que la entrega de premios suele estar planificada para aproximadamente dos horas después de finalizar la competición.

Para añadir más emoción al evento, la clasificación pública que pueden consultar los participantes y el público en general se congela una hora antes de finalizar la competición. De esta manera hasta la entrega de premios nadie sabe el ganador ni la posición en la que finalmente han quedado los equipos.

En cuanto el equipo de jueces da por verificada la clasificación final otro equipo de personas se encarga de imprimir los diplomas con las clasificaciones para que estén en el momento de iniciar la entrega de premios. Al principio de la ceremonia de entrega de premios se explican posibles estrategias de solución de los problemas. La Figura 1 muestra varios de los jueces explicando cómo solucionar uno de los problemas. La Figura 2 muestra el salón de actos momentos antes de la entrega de premios.

## 3. Organización

Organizar una competición de programación internacional en la que participan equipos de varias universidades implica varios aspectos adicionales con respecto a la organización de congresos. Uno de estos aspectos es el conjunto de problemas cuyos detalles se explican en la siguiente sección. Otro aspecto es la in-



Figura 2: Momentos antes de la entrega de premios en SWERC 2012.

infraestructura y la dependencia que de ella se tiene. La infraestructura incluye los ordenadores que utilizarán los equipos participantes durante las cinco horas de la competición más el sistema de jueces automáticos. Un fallo durante la competición puede ser motivo de suspensión y aplazamiento, lo que implicaría la preparación de un nuevo conjunto de problemas.

### 3.1. Configuración de los ordenadores

Los ordenadores que se utilizan como puestos de trabajo durante la competición son ordenadores de escritorio convencionales, los mismos utilizados en las prácticas de laboratorio pero configurados para no disponer de acceso a Internet salvo al juez automático. Su configuración es igual a la de los ordenadores que actúan como jueces automáticos, con la misma versión de los compiladores para los lenguajes permitidos. Estos ordenadores se instalan y configuran con el sistema operativo Linux, incluyendo los editores de programas y entornos de desarrollo integrado más utilizados y, por supuesto, los compiladores. También se instala en local toda la ayuda y documentación necesaria y permitida durante la competición, básicamente la *Java API Documentation*, los manuales en línea de la *C++ STL* y los de las librerías estándar de *C/C++*.

Todos los puestos de trabajo deben prepararse igual, es decir, deben ser réplicas de una única instalación. Para evitar periodos de espera excesivamente largos en caso de que falle un ordenador es mejor disponer de dos ordenadores por cada equipo participante, uno de ellos de reserva.

### 3.2. Juez automático

El sistema que actúa como juez automático se compone de un servidor más diez jueces. El servidor, al que acceden los participantes desde sus puestos (cada uno con su usuario y contraseña), contiene la Web que sirve de interfaz y la base de datos. Los diez jueces son ordenadores con la misma configuración excepto el servicio Web, pero sí contienen una réplica de la base de datos. Se configuran para solicitar continuamente tareas pendientes al servidor, por cada tarea reciben el código fuente, lo compilan y lo ejecutan contra todos los casos de prueba y envían el veredicto al servidor.

El sistema de comprobación automática, formado por estos once ordenadores más la infraestructura de la red, es el punto más crítico durante la competición. Un fallo durante las cinco horas puede ser un desastre. El personal técnico encargado de administrar los servidores pasa por una situación de estrés considerable, pues todo su trabajo de preparación realizado durante los meses anteriores depende de que no falle ningún componente.

El sistema de juez automático utilizado ha sido el *DOMjudge* [3], desarrollado en la Universidad de Utrecht. Es libre, gratuito y de código abierto.

### 3.3. Equipo humano

Aunque ya han sido citados los jueces humanos y los administradores del sistema, para que todo salga bien hace falta un gran equipo humano cuya colaboración es indispensable. El equipo humano se distribuyó

en distintas responsabilidades según muestra el Cuadro 2, varias personas estuvieron a cargo de más de una responsabilidad y/o colaboraron en varias tareas. Por ejemplo, los redactores de los problemas fueron también los jueces humanos, un equipo de personas que se reunió periódicamente desde unos cuatro meses antes de la competición para tener el conjunto de problemas bien definido y probados todos los casos de prueba. Los coordinadores del evento trabajaron en varias tareas, todas ellas preparadas también con varios meses de antelación, como la impresión de carteles y pósters, las camisetas, los diplomas acreditativos, los identificadores, las bolsas con regalos y material publicitario, las medallas y su diseño, etc. El registro y la vigilancia de las aulas donde se realizó la competición fueron las tareas que más personas involucraron mientras estuvieron activas. Otra tarea que requirió gran esfuerzo fue la preparación de todas las listas necesarias para tener identificados a alumnos competidores, profesores entrenadores, acompañantes, invitados, representantes de empresas patrocinadoras y los roles enumerados en el Cuadro 2.

Director competición
Jefe de jueces humanos
Jueces humanos
Redactores de los problemas
Administradores de sistemas
Coordinadores del evento
Responsables del registro
Alumnos voluntarios
Comité asesor

Cuadro 2: Distribución de responsabilidades.

## 4. Conjunto de problemas

El conjunto de problemas es quizás la parte más complicada de organizar en una competición de programación de carácter internacional. El nivel de los participantes varía, como es lógico, pero es bastante alto. De hecho, la dificultad media de los problemas supera con creces lo que se imparte habitualmente en las titulaciones de informática. Todo el que participe con intención de quedar entre los primeros puestos debe entrenar duro.

Nº problemas	Nivel de dificultad
2	Bajo
3	Medio
3	Alto
2	Muy alto

Cuadro 3: Niveles dificultad entre los diez problemas.

La dificultad de los problemas se distribuye según se indica en el Cuadro 3. El objetivo es que casi todos los equipos puedan resolver al menos un problema y al mismo tiempo evitar que ningún equipo acabe antes de las cinco horas. No se persigue que los resuelvan todos, sino que estén participando hasta el final. Los problemas de nivel bajo están pensados para que cualquier estudiante universitario medio sea capaz de resolverlos. Debería ser capaz cualquier alumno de tercer curso o superior. Los niveles medio y alto tienen por objeto ir clasificando a los equipos según capacidad de resolución de problemas y organización para el trabajo en equipo. Y como se ha comentado más arriba, el cometido de los dos más difíciles es evitar que ningún equipo acabe antes del tiempo reglamentario.

### 4.1. Tipología de los problemas

Los problemas que aparecen en este tipo de competiciones suelen versar sobre una o varias temáticas dentro de las Matemáticas, las Ciencias de la Computación y la Algorítmica. Excepto los dos más fáciles que, en muchos casos, admiten una solución simple o por fuerza bruta.

Ordenación
Algoritmos voraces
Simulación
Teoría de números
Combinatoria
Álgebra
Backtracking
Programación dinámica
Grafos
Geometría
Geometría computacional

Cuadro 4: Temáticas de los problemas.

El Cuadro 4 muestra la lista de temáticas que históricamente se utilizan en los problemas. Algunos de los problemas pueden requerir una solución que combine conocimientos de varias de estas temáticas. Aunque no figure como una temática, el uso de estructuras de datos avanzadas gestionadas eficientemente va implícito.

En algunos problemas es imprescindible utilizar técnicas algorítmicas avanzadas como son A\*, IDA\*, redes de flujo, etc. [9].

### 4.2. Entrenamiento

Los alumnos pueden entrenarse en distintos jueces automáticos activos las 24 horas. Los más utilizados son el *UVa Online Judge* de la Universidad de Valladolid [4], y la *USA Computing Olympiad* [5]. La ventaja

de este último es que cada usuario registrado va progresando individualmente subiendo niveles de dificultad. Para subir de nivel debe resolver todos los problemas del nivel anterior. Este sistema orienta al usuario mediante explicaciones sobre cómo abordar cada problema y, en caso de fallo, le muestra en qué casos de prueba falla su solución y cuál debería ser la salida (de especial interés para los principiantes). El otro juez, el de la UVa, reúne en su repositorio casi 3000 problemas y contiene los problemas de distintas ediciones de la ACM-ICPC, finales mundiales y regionales.

Existen competiciones periódicas en línea que sirven como entrenamiento y para prepararse las entrevistas de los procesos de selección. En cada vez más empresas tecnológicas de fuera de nuestro país se valora la participación en estas competiciones: TOPCODER [6], CODEFORCES [7] y Google code jam [8].

Además, con objeto de propiciar la participación en estas competiciones, la ETSInf de la UPV, al igual que la Facultad de Informática de la Universitat Politècnica de Catalunya, tiene una asignatura de libre elección que inicia a los alumnos en la participación en las competiciones. El temario de la asignatura en la ETSInf sigue los contenidos del libro *Programming Challenges* [9], que dispone de una página Web asociada para facilitar la labor del profesor y el aprendizaje de los alumnos.

## 5. Salida laboral

Al margen de ser una buena experiencia para el equipo organizador, en las circunstancias actuales deben considerarse las posibilidades de promoción profesional que participar en la ACM-ICPC ofrece a los alumnos de informática, tanto si llegan a la final mundial como si participan en sus regionales, la SWERC por ejemplo. Y también si participan en otras competiciones como se ha comentado en la sección anterior sobre entrenamiento.

La experiencia de haber participado en la SWERC desde 2003 es positiva, algunos exalumnos de la ETSInf han sido contratados por Google y otros han realizado estancias durante el verano. Pero no únicamente Google, de hecho es IBM quien patrocina la ACM-ICPC, corre con todos los gastos de la final mundial y aporta la mayor parte de los fondos para las regionales. En el caso de la SWERC 2012 también fueron patrocinadores facebook, CORITEL y TecSible (propietaria del producto DialApplet).

Como se ha comentado en el resumen y en la introducción, el objetivo de estas empresas es reclutar a los alumnos más brillantes que han demostrado su interés y su capacidad en la resolución de problemas complejos como son los que se plantean en este tipo de

competiciones. También Microsoft patrocina competiciones de programación con el mismo propósito.

## 6. Conclusiones

En primer lugar, queremos destacar aquí la satisfacción de haber organizado un evento para estudiantes de grado y máster. Algo poco habitual debido al esfuerzo que supone para un número considerable de profesores y voluntarios cuando no se traduce como un mérito en el currículum.

El esfuerzo, desde el punto de vista organizativo, es muy superior al de un congreso científico por dos motivos. En primer lugar, la preparación de la infraestructura para que todo funcione a la perfección (no es viable retrasar unos días algo para lo cual se han desplazado casi 200 personas). Este hecho implica un alto nivel de estrés durante el evento a diferencia de un congreso, en el cual, prácticamente tras el arranque, todo se puede relajar. En segundo lugar, la preparación del conjunto de problemas que los participantes deben resolver. Cada problema debe estar bien definido, probadas varias soluciones en distintos lenguajes de programación, y comprobados los casos de prueba. El mayor temor del equipo de jueces es dar por mala una buena solución. El segundo temor más importante es que falle el ordenador que hace de juez automático. Durante cinco horas todo depende de que este ordenador no falle.

Finalmente, queremos destacar el apoyo recibido desde la propia Universidad y desde las direcciones de la Escuela y el Departamento, tanto a nivel de infraestructura como a nivel económico.

## Referencias

- [1] ACM-ICPC <http://icpc.baylor.edu>
- [2] Southwestern Europe Regional Contest <http://swerc.eu>
- [3] DOMjudge <http://domjudge.sourceforge.net>
- [4] UVa Online Judge <http://uva.onlinejudge.org>
- [5] USA Computing Olympiad <http://www.usaco.org>  
Portal de entrenamiento: <http://ace.delos.com/usacogate>
- [6] TOPCODER <http://topcoder.com/tc>
- [7] CODEFORCES <http://codeforces.com>
- [8] Google code jam <http://code.google.com/codejam>
- [9] Steven S. Skiena, Miguel A. Revilla. *Programming Challenges: the programming contest training manual*. Springer, 2003