

ADA-Madrid



Relada

(Revista Electrónica de ADA)

Vol. 5 (1) 2011

ISSN: 1988-5822



Plataforma de Red Social para la Realización de Juegos

Arturo de Salabert. Fernando Díez.

Universidad Autónoma de Madrid.
Escuela Politécnica Superior. Dpto. de Ingeniería Informática.
arturo.desalabert@uam.es

Resumen: Este artículo describe una nueva plataforma P2P para la realización de enfrentamientos entre jugadores de forma interactiva en las prácticas de la asignatura Inteligencia Artificial, concretamente en el estudio de la Búsqueda entre Adversarios (Teoría de Juegos). Esta es la segunda de un conjunto de dos plataformas independientes técnica y funcionalmente, una para torneos y otra para enfrentamientos individuales, con características de red social. Ambas iniciativas han sido parcialmente financiadas con ayudas para el Desarrollo de las Enseñanzas de la Oficina de Convergencia Europea en los años 2009 y 2010. La primera ha sido utilizada intensivamente en los cursos 2009/10 y 2010/11 y la segunda ha sido probada con éxito por una selección de voluntarios en el curso 2010/11.

Palabras clave: Búsqueda entre adversarios. Inteligencia Artificial. Aprendizaje activo. P2P. Red Social. Auto aprendizaje. Virtualización del aprendizaje.

Abstract: This article introduces a new platform for executing P2P games in relation to the study of Adversarial Search (Game Theory), part of the practical curriculum in Artificial Intelligence. This is the second of two technically and functionally independent platforms: the first one for performing tournaments, and this second one for performing individual matches, with some social network characteristics. Both initiatives were partially funded by two grants from the European Convergence Office, in 2009 and 2010. Both platforms have been successfully used during the academic year.

Keywords: Adversarial search. Artificial Intelligence. Action learning. P2P. Social network. Self-learning. Learning virtualization.

INTELIGENCIA ARTIFICIAL Y TEORÍA DE JUEGOS

Contenido de la asignatura

La asignatura Inteligencia Artificial en la Escuela Politécnica Superior (en adelante EPS) de la Universidad Autónoma de Madrid es una asignatura troncal de segundo ciclo en la titulación de Ingeniería Informática así como de los Estudios Conjuntos en Ingeniería Informática y Matemáticas. Pertenece también, con carácter optativo, a la titulación de Ingeniería de Telecomunicación. El presente curso 2010-2011 ha sido el último en el que se ha impartido la asigna-

tura como parte del Plan de Estudios del año 1993 a extinguir (BOE 03/02/1993).

La asignatura se imparte en el primer semestre de cada curso académico. La matrícula del curso ronda los 200 estudiantes (221 en el presente), divididos en tres grupos de teoría y ocho grupos de prácticas. Las prácticas de la asignatura se efectúan en sesiones de laboratorio de dos horas semanales de duración. Por su parte, las clases de teoría se imparten en tres sesiones semanales de una hora de duración. Los contenidos de la asignatura abarcan los temas relacionados con Lógica (proposicional y de predicados), Agentes de Búsqueda (búsqueda no informada, búsqueda heurística y búsqueda entre adversarios) y Planificación. La evaluación se lleva a cabo mediante un examen parcial, efectuado en mitad del cuatrimestre, un examen final y las calificaciones obtenidas por los estudiantes en las prácticas de laboratorio.

En el plan de estudios de esta asignatura resulta de gran importancia su componente práctica para obtener una adecuada formación de los estudiantes en los contenidos de la misma. Las prácticas de la asignatura se realizan de manera continuada durante el cuatrimestre en los laboratorios de la Escuela. El trabajo que se desarrolla en ellos constituye una herramienta eficaz para entender y aplicar los conceptos y algoritmos presentados en las clases teóricas. El trabajo de los estudiantes en el laboratorio se desarrolla en lenguaje Lisp. Tiene una componente inicial de aprendizaje del paradigma de Programación Funcional, con el que tienen su primer contacto intensivo en esta asignatura. Las cuatro prácticas que realizan a lo largo del cuatrimestre tienen por objeto la profundización, mediante el desarrollo de software ad-hoc, en los conceptos vistos en las clases teóricas. Por tanto, en la primera práctica aprenden los conceptos básicos del lenguaje Lisp. En la segunda realizan algún desarrollo para la resolución de problemas de lógica proposicional; en la tercera aplican conocimientos de búsqueda heurística y, en la cuarta, se dedican al estudio de problemas de búsqueda entre adversarios. La búsqueda entre adversarios involucra el diseño de estrategias que permitan a un jugador automático ganar tanto a otros jugadores automáticos como, inclusive, a humanos. Una de las cosas que más sorprende y motiva a los estudiantes es verse capaces de diseñar estrategias que ganan a jugadores a los que ellos no consiguen ganar por mucho que se esfuercen.

Antecedentes de la experiencia realizada

De cara a la implantación del nuevo Grado en Informática, el equipo docente de la asignatura ha venido realizando en los últimos cursos académicos determinadas actividades encaminadas a la implantación de una metodología de trabajo acorde con las nuevas directrices sobre el desarrollo de destrezas y competencias del Plan Bolonia. En este sentido se han realizado distintas experiencias de trabajo colaborativo. Resumimos a continuación las experiencias más destacadas.

En el curso 2004-2005, haciendo uso de la plataforma KnowCat [Cobos, 2003] pusimos en marcha una metodología colaborativa de trabajo continuo en grupo a lo largo del cuatrimestre sobre el tema *El futuro: ¿humano, robot o ciborg?* [Díez, 2007]. Como parte de la evaluación se llevó a cabo una sesión final de debate entre todos los estudiantes de la asignatura. Dicha sesión de

debate fue grabada y se produjo, financiado por Vicerrectorado de Innovación de la UAM, un vídeo de una hora de duración, el cual se encuentra disponible en la Biblioteca¹ del centro como parte del material docente de la asignatura.

En el curso 2009-2010 se desarrolló una plataforma web para la realización de torneos [Salabert, 2010]. Dicha plataforma permitía la realización de partidas entre jugadores desarrollados por los estudiantes, los cuales mandan su código a un servidor. En dicho servidor se procesan los jugadores recibidos efectuando partidas entre ellos y generando una clasificación que se muestra on-line a los usuarios. Los jugadores, que son enviados por los alumnos a voluntad (aunque limitando las entregas por motivos explicados en el artículo), se enfrentan entre sí en una serie de partidas para establecer una clasificación, que actualiza cada hora, indicando el rendimiento de cada jugador. Esta plataforma ha sido utilizada masivamente tanto durante el curso pasado como en el actual, estando a disposición de los alumnos 24 horas al día durante un período de 4 semanas. En ambos años el total de estudiantes matriculados (unos 200) remitió al torneo más de 900 entregas de jugadores, las cuales generaron más de 100.000 enfrentamientos.

Un entorno similar en propósito al desarrollado para las prácticas de la asignatura es el denominado *Google AI Challenge*², desarrollado por el club de Informática de Universidad de Waterloo y financiado por la compañía Google.

PARTIDAS P2P

Motivación y antecedentes

La idea de realizar torneos entre estudiantes nace en el curso académico 2006-2007. La implicación de los estudiantes fue altísima y los resultados que obtuvimos fueron muy positivos. El único inconveniente fue la organización y desarrollo del torneo puesto que los enfrentamientos se realizaban de forma manual. Este hecho impedía la realimentación del estudiante y se perdía el efecto de superación y aprendizaje para la mejora.

Durante el curso 2008-2009 se realizó una simple clasificación de los jugadores off-line, con el limitado objetivo de calificar el código presentado por los alumnos. La única realimentación que recibían los estudiantes era su nota.

La experiencia llevada a cabo en el curso 2009-2010 cambió radicalmente la perspectiva de la utilidad de la actividad. Comprobar la implicación de algunos estudiantes en la actividad de programación de jugadores y el interés que mostraron por el torneo, nos animó a evolucionarlo hacia una plataforma más interactiva, con características de red social. Así pues, durante el curso 2010-2011, ante el éxito de la plataforma de juego en red y en respuesta a sugerencias realizadas por algunos alumnos, decidimos poner a punto un nuevo tipo de plataforma en la que los jugadores no se enfrentaran de un modo establecido por el árbitro del torneo, sino que pudieran enfrentarse unos a otros de manera voluntaria, selectiva e ilimitada.

La nueva plataforma debería permitir las partidas entre alumnos, mediante algún mecanismo de comunicación punto a punto. En la versión anterior el

¹ <http://biblos.uam.es/uhtbin/cgisirsi/IEoGn8BD9a/FILOSOFIA/79400091/123>

² <http://ai-contest.com/>

servidor del torneo realizaba las partidas seleccionando los contrincantes de cada jugador de forma aleatoria entre los de un conjunto. Sin embargo nos pareció natural, al margen del torneo oficial, crear un espacio donde fuesen los propios estudiantes quienes pudiesen retar a los contrincantes que considerasen oportunos, contra los que desean poner a prueba sus jugadores, al estilo de un club de ajedrez. De esta forma se reforzaría su aprendizaje y les permitiría enviar mejores jugadores al torneo. Para llevar a la práctica este nuevo esquema de juego decidimos crear una plataforma tipo red social para facilitar que los estudiantes pudiesen retarse unos a otros, así como sus historiales de partidas y resultados individuales. Puesto que esta nueva plataforma permite enfrentamientos punto a punto en tiempo real, la denominamos P2P (peer-to-peer).

Comparación entre la plataforma P2P y la de torneos.

Aunque ambas plataformas se están utilizando en la enseñanza de los mismos paradigmas de la Inteligencia Artificial, concretamente heurísticas y búsqueda entre adversarios [Russell & Norvig, 1995], las características y funcionalidades de la plataforma de torneos son bien distintas de la nueva plataforma P2P (tabla 1), por lo que esperamos utilizarlas simultáneamente en el próximo curso.

Característica	Torneo	Partidas P2P
Selección de contrincantes	Automática, mediante algoritmo de optimización	Red Social, libre decisión de ambos jugadores
Número de enfrentamientos	Limitado	Ilimitado
Nº de entregas	Limitado	Ilimitado
Ejecución	Centralizada en servidor	Distribuida, en la CPU local de cada contrincante
Proceso	Diferido	Tiempo real
Tamaño lote	Entregas recibidas en 60'	Individual
Tiempo de espera	5-60 minutos	1-3 segundos
Publicación	web	web
Tipo de Clasificación publicada	Puntos, función del número de victorias/derrotas	Flexible, según criterios de red social (popularidad, enfrentamientos, relaciones, puntos, ...)
Tipo de partida	Automática (entre algoritmos)	Automática (entre algoritmos), Mixta (humano vs algoritmo y Manual (entre humanos)
Evaluación	Según posición en clasificación	No evaluadas

Tabla 1. Plataforma de torneos vs plataforma P2P.

1. Requisitos de la Plataforma P2P

Los requisitos principales de la plataforma P2P fueron:

Requisito	Solución	Sts.
Confidencialidad del algoritmo de juego	Los jugadores comunican sólo jugadas, no programas	✓
Resistencia a errores de los clientes	El servidor es resistente a errores esperables de los clientes. Los clientes son robustos tanto frente a sus propios errores como a los del contrario	✓
Disponibilidad	El sistema permite la redundancia de servidores.	✓
Escalabilidad	Los procesos consumen la mayoría de recursos de los clientes, no del servidor, por lo que el sistema de publicación y control escala linealmente con un mínimo overhead por usuario	✓
Supervisión en tiempo real del estado del sistema	El challenger informa periódicamente al servidor del estado de la partida, que es hecho público inmediatamente en la web	✓
Independencia del tipo de juego	El sistema P2P está desacoplado y es independiente del juego	✓
Independencia del lenguaje y del SO	Sistema distribuido con comunicación entre procesos basado en comandos independientes de lenguaje y SO	✓
Compromiso de permanencia	El challenger se compromete a cierto número de partidas	✓
Limitación de imposturas	Basada en el control de IP's	Pdte.
Confidencialidad de las IPs	La red social está basada en alias únicamente	Pdte.
Posibilidad de revisar las partidas	El servidor almacena las jugadas, que pueden ser consultadas con ciertas limitaciones	✓
Disponibilidad 24x7	Configuración hardware y software de robustez industrial	✓
Posibilidad de juegos con múltiples jugadores	El esquema permite juegos con cualquier número de jugadores, siempre que se disponga de un algoritmo de secuenciación de jugadas	50%

Tabla 2. Requisitos de la plataforma P2P.

2. Implementación de la Plataforma P2P

La plataforma de juegos debería funcionar facilitando la interacción entre los estudiantes, pero sin contar con un carácter obligatorio. A los estudiantes se les pide entregar el jugador más eficiente que sean capaces de producir, el cual debe demostrar su capacidad en un torneo frente a todos los demás. Sin embargo, la plataforma P2P ofrece un espacio de experimentación selectiva, cuyos resultados ayudan, pero no afectan al torneo; algo así como un club de ajedrez donde los miembros experimentan, compiten, aprenden e interactúan al margen de su participación en torneos oficiales.

Puesto que la participación es voluntaria, y dado que la localización física de los estudiantes puede ser variable, el servicio principal ofrecido por la plataforma tiene que ser identificar a aquellos que declaran estar dispuestos a enfrentarse a contrincantes, a los que llamaremos *campeones*, de modo que el resto sepa cómo interactuar con ellos. Los interesados, a los que llamaremos *aspirantes*, pueden solicitar jugar partidas

con algunos de los *campeones* disponibles, quienes, a su vez, pueden aceptar o no el desafío, ya que no están obligados a tomar parte en todas las partidas solicitadas. Como se verá el procedimiento es idéntico al de las redes sociales habituales, donde el atributo *amistad* es sustituido por *partida*. Los roles campeón y aspirante son voluntarios y mutuamente compatibles. Cualquier estudiante puede participar en uno o en ambos roles simultáneamente. De hecho puede hacerlo múltiples veces y con alias distintos, y si no la revela, nadie (salvo el administrador), conocerá la identidad del jugador.

Confidencialidad del algoritmo

La tabla de requisitos de la plataforma P2P (tabla 2) está ordenada según la prioridad relativa de los mismos. El primero de ellos es garantizar la confidencialidad del algoritmo de juego. Se trata de que dos jugadores puedan enfrentarse en tiempo real, estando situados en lugares remotos sin necesidad de intercambiar código con el contrario. La comunicación entre los jugadores es punto a punto y basada en el intercambio de jugadas. Cada uno de los jugadores ejecuta su algoritmo de juego en local y espera recibir la jugada del contrario reduciendo al mínimo tanto las interacciones entre jugadores como entre ellos y el sistema de información.

Tolerancia a errores

La robustez frente a errores de los clientes implica controlar tres aspectos: la robustez de cada cliente frente a los errores propios y a los del contrario, y la robustez del servidor frente a los errores de los clientes. En todos los casos, los errores pueden ser tanto involuntarios como malintencionados. Frente a errores involuntarios de cualquiera de los componentes, la respuesta del resto es controlada, ya sea detectando y registrando el error o deteniendo su ejecución ordenadamente.

En la plataforma de torneos, salvo el código que contiene el algoritmo con la estrategia del jugador, todo el código reside en el servidor o servidores del torneo. La interacción del alumno con el sistema se reduce a remitir mediante una interfaz web el código de su jugador y a leer la clasificación en otra página. Aparte de los errores propios del servidor, el asegurar la robustez del sistema se reduce a analizar el código entregado por el alumno. Por el contrario, en la plataforma P2P el código ejecuta en al menos tres lugares distintos, de los cuales al menos dos (los clientes) corren riesgo de manipulación malintencionada. A diferencia del torneo, donde lo que los alumnos entregan son programas que tienen que ejecutar en el mismo servidor que controla el torneo, con el consiguiente riesgo de troyanos, virus o inyecciones de código, la seguridad en la plataforma P2P es bastante más sencilla de garantizar, pues lo único que se intercambia entre contrincantes son jugadas. A fin de asegurar que la jugada realizada por el contrario es correcta, cada jugador reproduce en su máquina local tanto las jugadas propias como las del contrario. En caso de discrepancias se avisa al servidor, que puede automáticamente descalificar jugadores.

Asimismo, la plataforma P2P es resistente a caídas del sistema, pues es capaz de recuperar su estado anterior a la caída, el cual se almacena cada po-

cos segundos. Durante el periodo de pruebas del sistema P2P surgieron problemas o sugerencias de mejora que se fueron incorporando según aparecían. Ello se realizó en tiempo real y de un modo transparente a los usuarios gracias tanto a la capacidad de Lisp de modificar partes de código sin detener la ejecución como a la coexistencia de un sistema redundante basado en una versión de producción más otra de pruebas ejecutando concurrentemente.

Disponibilidad y escalabilidad

Otros requisitos fundamentales del sistema son su disponibilidad y escalabilidad. La disponibilidad está garantizada permitiendo la redundancia de servidores. Por su parte, la escalabilidad está garantizada reduciendo al mínimo las interacciones entre el servidor y los contrincantes. La comunicación del servidor es exclusivamente con los campeones. Los aspirantes no tienen ningún contacto con el servidor. Esto puede parecer irrelevante en juegos uno contra uno (reduciría a lo sumo al 50% el número de interacciones), pero es fundamental en juegos de grupo con múltiples participantes, donde puede representar diferencias de varios órdenes de magnitud. Por otra parte, tanto la frecuencia de las interacciones entre servidor y campeones como el tamaño de los paquetes intercambiados son mínimos. Extrapolando los resultados de pruebas realizadas con hasta 8 partidas simultáneas, podemos asegurar que los 200 alumnos podrían estar todos jugando (100 partidas simultáneas) sin consumir ni el 10% de los recursos del servidor, aunque para ello habría que reducir la frecuencia de refresco de la publicación web (lo cual no tendría ningún efecto perceptible).

Supervisión del estado del sistema

La comunicación periódica entre servidor y clientes permite disponer de información en tiempo real del estado del sistema. La Fig.1 muestra un detalle de la información publicada.

Independencia del juego

El desacoplamiento de las funciones del juego y las de comunicaciones permite que la plataforma P2P acepte no sólo cualquier tipo de juego, sino que podría gestionar varios juegos distintos simultáneamente.

Multiplataforma

La parte servidor de la plataforma P2P, aunque programada íntegramente en Lisp, es independiente del sistema operativo y del lenguaje de programación utilizado por los clientes. El único requisito de éstos es la comunicación mediante Berkeley sockets con el servidor. No se han realizado pruebas con diferentes lenguajes, pero sí con diferentes SO's. Durante la fase de pruebas, el servidor ha ejecutado tanto sobre plataformas Windows como Red Hat Enterprise Linux y los clientes han estado ejecutando sobre diferentes versiones de Windows, Linux y MacOS, todas ellas simultáneamente.

Fecha Ult. Act.: 11-01-21 22:06:32 [ms]		Notas, avisos y comentarios		FAQ's P2P				
CHULOS (Disponibles Ocupados)								
Nombre	Stat	Vers	IP	Puerto	Acepta Actual	Result (E V D)	Contrincantes (aspirante/resultado) [0=Emp,1=Vict,2=Der]	
Jug-Mmx-Bueno@TABLET-PARADISE	new	5.2	81.32.145.124	9000	88	NIL	#(0 30 0)	Humano@ANTRAZIT 5.1 14/1, Humano@DESPACHO 5.1 46/1, Humano@DESPACHO 5.1 14/1, Humano@DESPACHO 5.1 14/1, Humano@DESPACHO 5.1 14/1, Jug-Mmx-Bueno@ALVARO 5.1 56/1, Humano@ALVARO 5.1 24/1, Jug-Mmx-Bueno@TABLET-PARADISE 5.1 56/1, Humano@VICENTECALDERON 5.1 28/1, Jug-Mmx-Bueno@TABLET-PARADISE 5.1 56/1, JUGADOR-ALEATORIO@VICENTECALDERON 5.1 28/1, JUGADOR-ALEATORIO@VICENTECALDERON 5.1 24/1, JUGADOR-ERRONEO@TABLET-PARADISE (Error func.(abandono) 5.1 1/1, JUG-MMX-REGULAR@TABLET-PARADISE 5.1 96/1, JUG-MMX-REGULAR@TABLET-PARADISE 5.1 96/1, Humano@TABLET-PARADISE 5.2 16/1, JUG-MMX-REGULAR@TABLET-PARADISE 5.2 96/1, Humano@ALVARO 5.2 24/1, JUGADOR-ALEATORIO@VICENTECALDERON 5.2 20/1, JUG-MMX-REGULAR@VICENTECALDERON 5.2 96/1, Jug-Mmx-Bueno@TABLET-PARADISE 5.2 56/1, JUGADOR-ALEATORIO@TABLET-PARADISE 5.2 18/1, JUGADOR-ALEATORIO@TABLET-PARADISE 5.2 20/1, Jug-Mmx-Bueno@VICENTECALDERON 5.2 56/1, Humano@HCC-PC 5.2 14/1, Humano@HCC-PC (Fatal: NIL) 5.2 11/1, Humano@HCC-PC 5.2 12/1, Humano@DESPACHO 5.2 14/1, Humano@DESPACHO 5.1 14/1, Humano@DESPACHO 5.1 14/1, Humano@DESPACHO 5.1 26/1, Humano@DESPACHO 5.1 28/1, JUGADOR-ALEATORIO@ANTRAZIT 5.1 8/1, Jug-Mmx-Bueno@ALVARO 5.1 56/1, Humano@ALVARO 5.1 38/1, Humano@HCC-PC wrong vers: 5.2, Humano@HCC-PC wrong vers: 5.2, Jug-Mmx-Bueno@VICENTECALDERON wrong vers: 5.2, Jug-Mmx-Bueno@ALVARO wrong vers: 5.2,
Jug-Mmx-Bueno@SOBREMESA	rdy	5.1	95.22.182.245	9000	100	NIL	#(0 11 0)	Humano@DESPACHO 5.1 14/1, Humano@DESPACHO 5.1 14/1, Humano@DESPACHO 5.1 26/1, Humano@DESPACHO 5.1 28/1, JUGADOR-ALEATORIO@ANTRAZIT 5.1 8/1, Jug-Mmx-Bueno@ALVARO 5.1 56/1, Humano@ALVARO 5.1 28/1, Humano@HCC-PC wrong vers: 5.2, Humano@HCC-PC wrong vers: 5.2, Jug-Mmx-Bueno@VICENTECALDERON wrong vers: 5.2, Jug-Mmx-Bueno@ALVARO wrong vers: 5.2,
Jug-Mmx-Bueno@ALVARO	rdy	5.2	213.37.36.215	9000	100	JUG-MMX-REGULAR@TABLET-PARADISE	#(0 11 0)	Jug-Mmx-Bueno@ALVARO 5.1 56/1, Humano@ALVARO 5.1 28/1,
Jug-Mmx-Bueno@SAGA	idl	5.2	150.244.58.94	9000	81	NIL	#(0 9 0)	Humano@SAGA 5.2 14/1, Humano@ANTRAZIT Incompatible ver:5.1, Humano@DESPACHO Incompatible ver:5.1, Humano@DESPACHO Incompatible ver:5.1, Humano@DESPACHO Incompatible ver:5.1, Humano@SAGA (Error func.(abandono) 5.2 1/1, Humano@ALVARO Incompatible ver:5.1, Humano@ALVARO 5.2 34/1, Humano@HCC-PC 5.2 28/1, JUG-MMX-REGULAR@VICENTECALDERON (Fatal: NIL) 5.2 1/1, Humano@HCC-PC 5.2 40/1, Jug-Mmx-Bueno@VICENTECALDERON (Fatal: NIL) 5.2 1/1, Humano@DESPACHO 5.2 14/1, Humano@DESPACHO 5.2 16/1, JUGADOR-ALEATORIO@VICENTECALDERON 5.1 28/1, JUG-MMX-REGULAR@VICENTECALDERON 5.1 96/1, JUGADOR-ALEATORIO@VICENTECALDERON 5.2 20/1, Jug-Mmx-Bueno@VICENTECALDERON 5.2 56/1,
Jug-Mmx-Bueno@VICENTECALDERON	rdy	5.2	83.54.214.244	9000	100	NIL	#(0 13 0)	JUGADOR-ALEATORIO@VICENTECALDERON 5.1 28/1, JUG-MMX-REGULAR@VICENTECALDERON 5.1 96/1, JUGADOR-ALEATORIO@VICENTECALDERON 5.2 20/1, Jug-Mmx-Bueno@VICENTECALDERON 5.2 56/1,
Jug-Mmx-Bueno@JORGEPC	idl	5.1	81.35.33.46	9000	100	NIL	#(0 11 0)	

DESCALIFICADOS (pueden volver a jugar presentando un jugador correcto)
Nombre Motivo

Figura 2. Estado del torneo.

CONCLUSIONES

Tras el éxito del formato de torneo, la posibilidad de realizar partidas uno a uno ha despertado entre los estudiantes una enorme expectación. Muchos han estado interesándose por la disponibilidad de la plataforma desde el mismo momento en que tuvieron constancia de su existencia. La puesta en marcha del proyecto se tuvo que retrasar varias semanas (por lo cual sólo fue posible realizar una fase de pruebas con un grupo limitado). Sin embargo, pese a que dichas pruebas coincidieron con la semana de exámenes, muchos de los voluntarios pusieron sus ordenadores privados toda una semana a disposición de los participantes y estuvieron hasta 20 horas diarias pendientes de la evolución de los encuentros, remitiendo múltiples comentarios y sugerencias. Todo ello es gratamente sorprendente, más cuando las pruebas se realizaron después del torneo, con lo que los participantes no pudieron obtener ventajas prácticas que aplicar a sus jugadores pese al considerable tiempo invertido. Varios de ellos se ofrecieron para cualquier otro tipo de trabajo relacionado con el tema.

En resumen, podemos afirmar que el potencial de este tipo de iniciativas es inmenso, ya que no sólo estimulan el interés de los estudiantes por la experimentación y el aprendizaje, sino que permiten galvanizar un nivel de participación desconocido en otros ámbitos.

BIBLIOGRAFÍA

Cobos, R. (2003), Mechanisms for the Crystallization of Knowledge, a proposal using a collaborative system. Doctoral Thesis. Universidad Autónoma de Madrid, Spain, 2003.

de Salabert, A., Suárez, A., Cobos, R., Díez, F. (2010) Plataforma en Red para la Realización de Torneos y Trabajo Colaborativo. Relada 4(4): 314-321.

Díez, F. and Cobos, R. (2007), A case study of a cooperative learning experiment in artificial intelligence. Computer Applications in Engineering Education, 15: 308–316. doi: 10.1002/cae.20114.

Russell S, Norvig P (1995) Artificial Intelligence: A Modern Approach, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey.

Recibido: 11 marzo 2011.

Aceptado: 11 abril 2011.