

**ADA-Madrid**



# Relada

(Revista Electrónica de ADA)

**Vol. 4 (4) 2010**

ISSN: 1988-5822



## Plataforma en red para la realización de torneos y trabajo colaborativo

**Arturo de Salabert**  
**Alberto Suárez**  
**Ruth Cobos**  
**Fernando Díez**

Universidad Autónoma de Madrid.  
Escuela Politécnica Superior. Dpto. de Ingeniería Informática.  
[arturo.desalabert@uam.es](mailto:arturo.desalabert@uam.es)

**Resumen:** Se describe una plataforma para la realización de torneos en relación con el estudio de Teoría de Juegos y para el trabajo colaborativo dentro de la asignatura de Inteligencia Artificial. La puesta a punto de la plataforma fue parcialmente financiada con una ayuda para el Desarrollo de las Enseñanzas de la Oficina de Convergencia Europea en 2009 y ha sido utilizada con éxito en el curso 2009/10.

**Palabras clave:** Búsqueda entre adversarios. Inteligencia Artificial. Aprendizaje activo. Entorno colaborativo. Auto aprendizaje. Virtualización del aprendizaje.

**Abstract:** A platform for performing tournaments, in relation to the study of Game Theory and for collaborative work in the study of Artificial Intelligence, is introduced. The platform was successfully used during the academic year 2009/10. Its implementation was partially funded by a grant from the European Convergence Office in 2009.

**Keywords:** Adversarial search. Artificial Intelligence. Action learning. Collaborative environment. Self-learning. Learning virtualization.

### INTRODUCCIÓN

#### **Contenido de la asignatura: Inteligencia artificial y teoría de juegos**

La asignatura de Inteligencia Artificial (Russell y Norvig, 1995) es una asignatura troncal de segundo ciclo de la titulación de grado en Ingeniería Informática y optativa para los estudios de Ingeniería de Telecomunicación en la Escuela Politécnica Superior (EPS) de la Universidad Autónoma de Madrid (UAM). Como parte de la adecuación de la asignatura a las nuevas metodologías de enseñanza, se está trabajando en la virtualización de la misma, favoreciendo la participación en grupo, la adaptación al alumno, el estímulo de su iniciativa (se trata de un problema abierto) y la exploración de recursos externos. Se imparte en el primer semestre y tiene una matrícula aproximada de 200 estudiantes, divididos en tres grupos de teoría y ocho de

prácticas. Además, uno de los grupos de teoría se imparte en inglés desde el curso académico 2004/05.

Los contenidos de la asignatura tratan de aspectos relacionados fundamentalmente con:

- Lógica
  - ✓ Lógica proposicional
  - ✓ Lógica de predicados
- Agentes de búsqueda
  - ✓ Búsqueda no informada
  - ✓ Búsqueda heurística
  - ✓ Búsqueda entre adversarios
- Planificación

### **Componente práctica de la asignatura**

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. Dicha componente consiste tanto en la realización de trabajos prácticos en el laboratorio (llamadas prácticas de laboratorio) como la realización de ejercicios teóricos específicos de la asignatura. Dichos ejercicios pueden consistir tanto en partes del código software que se pide implementar a los alumnos en el laboratorio, como ejercicios de aplicación de los conocimientos aprendidos en la asignatura basados en casos reales. Ejemplos de esto último pueden ser demostrar un teorema matemático mediante lógica, resolver un crucigrama mediante alguna estrategia de búsqueda heurística o resolver un problema de transporte de mercancías mediante planificación.

Las prácticas de la asignatura que se realizan de manera continuada durante el curso, son una herramienta eficaz para entender y aplicar los conceptos y algoritmos presentados en las clases teóricas. Una parte significativa de la asignatura está dedicada a problemas relacionados con la Teoría de Juegos (Samuel, 1959 y Turing, 1950) y, en concreto, al diseño de estrategias inteligentes para juegos, es decir, el desarrollo de jugadores automáticos cuyas estrategias para un determinado juego (damas, ajedrez, otello, go, etc.) son diseñadas por los estudiantes.

La práctica involucra el diseño de estrategias que permitan a un jugador automático ganar tanto a otros jugadores automáticos como a humanos. Recordemos que esta meta es asequible con la capacidad de cómputo de un ordenador de mesa. De hecho, 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.

## **UN TORNEO ON-LINE COMO ESTÍMULO AL APRENDIZAJE**

### **Motivación y antecedentes**

Inspirados en los principios del aprendizaje activo (*action learning*) (Samuel, 1959 y Turing, 1950), nos pareció que una forma natural para evaluar

de calidad de las estrategias automáticas diseñadas por los estudiantes podría ser la organización de un torneo entre los jugadores que utilizan dichas estrategias. Con el fin de permitir que éstos puedan determinar si las estrategias (algoritmos Lisp) diseñadas son efectivas, se ha puesto a punto un sistema de gestión de torneos al que los estudiantes pueden enviar sus jugadores y enfrentarlos entre ellos para contrastar su efectividad.

La idea de realizar un torneo en el que se celebraban partidas entre los jugadores automáticos desarrollados por los estudiantes se puso en práctica de forma experimental en el curso académico 2006/07. En aquella ocasión los resultados fueron muy positivos desde el punto de vista de la implicación de los estudiantes, que estaban muy motivados por dicho torneo. El único inconveniente fue la dificultad para organizarlo. Además, puesto que los enfrentamientos se realizaban de forma manual una vez entregada la práctica, se carecía de la dimensión de realimentación y del efecto de estímulo para la auto superación y el aprendizaje.

### Creación de la plataforma. Requisitos

Con la experiencia del proyecto anterior, durante el curso 2009-2010 decidimos la implementación de una primera versión de la plataforma en la que los jugadores se enfrentan entre sí en una serie de partidas para establecer una clasificación que se publica periódicamente poco después de las entregas.

Los requisitos que se consideraron esenciales en la plataforma fueron:

Requisito	Solución	Cumpl.
Recepción de jugadores desacoplada del torneo	A través del sistema estándar de entregas de la EPS	✓
Independencia del tipo de juego	El mayordomo o sistema de gestión del torneo es independiente del motor de juego, más aún, podría controlar simultáneamente más de un juego	✓
Limitación en nº de entregas	Mediante codificación de fecha y nº de entrega en el nombre del fichero	✓
Limitación de imposturas o falsas entregas	Mediante contraseñas individuales	✓
Limitación de abusos o entregas fuera de plazo.	Control mediante la fecha declarada y la fecha del sistema	✓
Seguridad contra ataques maliciosos al sistema	Analizador gramatical ( <i>parser</i> ) para inspección automática del código. Limitación de las funciones utilizadas y de accesos al SO. Desacoplamiento del servidor de torneos de los servidores de entrega y publicaciones	✓
Posibilidad de reproducir los enfrentamientos	El motor permite determinar la ventana de eventos a procesar, lo cual permite la reproducción de cualquier situación pasada o futura	✓
Supervisión en tiempo real del estado del sistema	Traza con marca horaria con precisión de segundos accesible on-line	✓
Disponibilidad 24x7	Configuración hardware y software de robustez industrial	✓
Incorporación de mejoras en tiempo real	El sistema de producción y el de pruebas son concurrentes e intercambiables	✓

Clasificación con un n <sup>o</sup> pequeño de enfrentamientos	Sistema de puntuación inspirado en el del ajedrez, donde la puntuación obtenida en una partida es función de las respectivas posiciones de los jugadores	✓
Opción de juegos colaborativos	Visibilidad, para grupos de tamaño y composición configurables, del código del jugador	Pdte.
Interacción con Moodle	Pendiente de evaluación	Pdte.
Publicación de resultados desacoplada del torneo	A través de la página web de la asignatura	✓

## Creación de la plataforma. Implementación

El primer requisito del sistema es garantizar la puntualidad, la autenticidad y la no repudiabilidad de las entregas. Ello se consigue mediante un sistema de codificación de nombres y de contraseñas personales e intransferibles.

El sistema de gestión del torneo tiene que ser capaz de gestionar cualquier tipo de juego. Ello se consigue mediante el desacoplamiento de las funciones de la mecánica del juego y las funciones de gestión de entregas, selección de contrincantes, puntuación de las partidas, preparación de la clasificación y publicación de resultados. El sistema de gestión implementado, (mayordomo), no sólo realiza las anteriores funciones independientemente del juego, sino que podría gestionar varios juegos simultáneamente.

Dado que lo que los alumnos entregan son programas que se tienen que ejecutar en el mismo servidor que controla el torneo, la seguridad es un requisito fundamental. Para ello se utiliza un analizador sintáctico o parser que se ha desarrollado independientemente para la evaluación de trabajos prácticos de la asignatura. Dicho parser inspecciona el código entregado antes de ejecutarlo y permite detectar tanto errores de formato como el uso indebido de funciones.

Los resultados de la clasificación final en el torneo van asociados a una nota. Eso hace necesario poder demostrar y justificar la corrección del proceso, tanto frente a posibles impugnaciones por los alumnos como contra caídas del sistema ya sean por errores del software como del hardware o de la infraestructura. De hecho el día 1 de enero de 2010 se dieron a la vez tres errores inesperados: caída del sistema central de entrega de prácticas, caducidad de la versión del intérprete (software gratuito) con la consecuente parada del motor del juego y caducidad de los permisos de acceso al sistema del autor de este artículo. En esas fechas no había nadie supervisando el torneo, pero gracias a los sistemas de traza fue posible analizar la secuencia de eventos y reiniciar el torneo de forma transparente. Aún así los alumnos reclamaron los días en los que no pudieron hacer entregas, lo cual refleja la importancia que otorgaban al torneo, pese a su pequeña influencia en la nota final y a que los dos días de falta de disponibilidad poco representaban frente a las 4 semanas que duró el torneo.

Durante el periodo del torneo surgieron problemas o sugerencias de mejora que se fueron incorporando según aparecían. Ello se realizó de un modo transparente a los usuarios gracias a la coexistencia de un sistema redundante basado en una versión de producción más otras dos de pruebas ejecutando concurrentemente en dos servidores distintos.

Otro requisito fundamental para mantener el atractivo de un torneo on-line es el presentar los resultados periódicamente y sin demasiado retardo frente a las entregas de jugadores. Por una parte, la única forma de garantizar que cada jugador ocupa el lugar que le corresponde en la clasificación es enfrentar todos contra todos. Este no es un problema trivial: hay un continuo flujo de jugadores que entran y que abandonan el juego y el enfrentar todos contra todos los jugadores activos antes de cada publicación de la clasificación es impracticable por la cantidad de computación requerida. El problema crece de forma geométrica con el número de jugadores (según  $N^2$  para  $N$  jugadores). A fin de buscar una solución suficientemente simple y a la vez aceptable, buscamos inspiración en la puntuación del ajedrez. Hemos implementado varias funciones de puntuación incluyendo incrementos tanto fijos como proporcionales a las puntuaciones de ambos contendientes y simulado su evolución a través de decenas de miles de partidas simuladas. Las simulaciones demostraron que enfrentando cada nuevo jugador contra jugadores veteranos elegidos convenientemente de entre los activos en ese momento, era posible tras unas 40 partidas situarlos en la clasificación en una posición que de media no distaba más de un 5% de la posición de los correspondientes. La metodología utilizada en este estudio y las conclusiones obtenidas merecen un estudio aparte, pero baste decir que la solución elegida soluciona el problema de modo lineal con el nº de nuevos jugadores e independientemente del nº de jugadores en activo.

Existe además una dimensión educativa del tiempo de refresco que se analiza en detalle en el siguiente apartado.

Finalmente, sería deseable la interacción con Moodle (<http://moodle.org/>), plataforma de virtualización de la enseñanza en la UAM. Conviene indicar que el tipo de problema resuelto con el torneo, así como en general la evaluación de código de libre desarrollo, exceden las capacidades de Moodle. Sería por tanto conveniente analizar la capacidad de Moodle de enlazar con sistemas externos, ya sea mediante API's o mediante interfaces de algún tipo.

### **Tiempo de refresco**

El tiempo de desfase máximo entre las entregas de los alumnos y la publicación de resultados es ajustable. En el torneo realizado en el curso 2009/2010 dicho desfase se estableció en 1 hora. Este tiempo no está limitado por cuestiones técnicas sino educativas. Para el juego actualmente implementado, una versión modificada de Mancala, la realización de los enfrentamientos necesarios para incorporar al torneo un nuevo lote de unos 10 jugadores podría realizarse, con la capacidad de cómputo de un servidor dedicado de gama media, en unos 50 segundos, con lo cual, si el flujo de llegada de nuevos jugadores no superara los 10 por minuto, podría actualizarse la clasificación publicada en la web una vez por minuto. Obviamente esos valores dependen sensiblemente del juego en cuestión. Juegos en los que haya más opciones requerirán mucho más tiempo de proceso, pero, como se explica a continuación, esa no es nuestra principal preocupación.

Nuestra experiencia demuestra que los alumnos de informática tienden a delegar una gran parte de su trabajo intelectual al ordenador. A diferencia de épocas anteriores, en las que ya fuera por la dificultad de obtener capacidad de

cómputo o por una actitud mental diferente, el trabajo de reflexión previo era mucho más importante que la interacción con la máquina, la actual ubicuidad y gratuidad de la computación crean el riesgo de invitar a un estilo de programación basado en un permanente proceso de prueba y error, donde la interacción con la máquina se realiza casi sin reflexión previa.

Creemos que el riesgo es significativo, comparable al de una persona que no fuera capaz de escribir una frase correcta sin un corrector, no sintáctico o de estilo, sino semántico (cosa que todavía no existe comercialmente). En otras palabras: el usuario mandaría “frases” a la máquina confiando que esta le confirme si la frase dice lo que pretende. Parafraseando a Wittgenstein (Wittgenstein, 1987), si uno sólo es capaz de decir aquello que entiende, con mayor razón, no debería ser posible programar lo que uno no ha pensado correctamente. Por tal motivo, es decir, para forzar a la reflexión y evitar la mera interacción lúdica con el ordenador, en el torneo se decidió limitar a 3 las entregas de jugadores (i.e. programas o algoritmos que implementan la función del jugador) por persona y día. Con un máximo de 3 entregas por día, obtener actualizaciones de la clasificación cada hora parecía más que suficiente.

### **Clasificación, estado del sistema, vista del profesor**

La Figura 1 muestra un detalle de la clasificación publicada periódicamente, en su versión del profesor. La clasificación vista por los alumnos contiene una visión reducida de la misma.

## **CONCLUSIONES**

El formato de torneo ha despertado gran interés entre los estudiantes. El 95% de los estudiantes matriculados en la asignatura ha participado en la experiencia mostrando, al menos de forma verbal, su satisfacción con la misma. En concreto, se ha observado una notable mejora en la calidad de las estrategias diseñadas respecto a años anteriores, en los que los estudiantes no disponían de un banco de pruebas en el que determinar la eficacia de los diseños realizados. Tras tres años de desarrollo de jugadores, por primera vez hubo alumnos que superaron los baremos considerados de excelencia. Más del 5% de los alumnos consiguieron este nivel, llegando su nivel de implicación a niveles desconocidos anteriormente, como demuestra el hecho de que 180 alumnos agrupados en 91 parejas remitieran más de 900 entregas. Ello implica una media de 10 entregas por pareja, aunque el nivel de entusiasmo no es homogéneo: algunas parejas se implicaron tanto en el torneo que llegaron a remitir hasta 30 jugadores mientras que otras sólo presentaron 1. En general puede afirmarse que la mayoría consideró la experiencia interesante, pues el 67% de las parejas entregaron 8 o más jugadores, mientras que sólo el 33% entregaron 7 o menos. Por ello hemos acordado continuar mejorando la metodología puesta en práctica, dotándola de más y mejores prestaciones.

Fecha Ult. Act.: 10-01-15 14:11:54 [Notas, avisos y comentarios](#) [FAQ's](#)

Enfrentamientos totales: 17150

Posicion	Nombre	Fecha	Alias	Puntos	Partidas	Funciones
1	TA062	100109	toy-2	36893/37	100	(NOCERO CAPTURA-CEDE DIF-FICHAS MI-F-EV)
2	MC041	100110	DL_LZ1	36677/37	100	(MI-F-EV FICH-PARES-NO-CERO POS-PARES-NO-CERO MEJOR-ROBO NO-AMBOS-PAI)
3	TD051	100113	-----	991	100	(DIFERENCIA-AMENAZADAS CUENTA-PROTEGIDAS GET-FICHAS-COLUMNA CUANT)
4	MD103	100115	Kostadin2	977	100	(FICHAS-PERDIDAS MINIMO_FICHA MAXIMO_FICHA SIGUE-GANANDO-O-PERDIEND
5	MD101	100114	Kostadin	106376/111	102	(FICHAS-PERDIDAS MINIMO_FICHA MAXIMO_FICHA SIGUE-GANANDO-O-PERDIEND
6	TD082	100113	SAWYER	104749/111	100	(ES-ACCESIBLE CUENTA-PAREJAS-ACCESIBLES MI-PAR CUENTA-CEROS DIFEREN
7	MD022	100108	Bambi	34689/37	98	(COMER ANALISIS MI-F-EV)
8	MD061	100113	Leyer-1	103762/111	100	(HOYO-CONTRARIO PAR CALCULA1 EVALUA-HOYO1 MI-F-EV)
9	MB091	100112	rebel_jr	103282/111	100	(OTRO-POSIBILIDAD MAX-L MI-F-EV)
10	AA013	091228	BlckKnight	920	100	(MI-F-EV)
11	MD093	100115	Vegeta	916	98	(NUEVA-WIKI MI-F-EV-MOVILIDAD MI-F-EV-NF MI-F-EV)
12	TC022	100110	Olisadebe	33850/37	100	(MI-F-EV)
13	TD053	100111	ElQTeFoca	33378/37	102	(GET-FICHAS-COLUMNA CUENTA-CAPTURAS CUANTAS-COMB-V3 CUANTAS-COMI
14	MD131	100115		902	100	(EVALUACION-CUBILETES ALCANZABLE NUM-ALCANZABLES MI-F-EV)
15	MD102	100110	PuloABambi	33357/37	98	(FICHAS-PERDIDAS MINIMO_FICHA MAXIMO_FICHA SIGUE-GANANDO-O-PERDIEND
16	TD133	100108		33316/37	100	(COMBO AMENAZA-A-POSICION NOS-AMENAZAN AMENAZAMOS GET-TABLERO M
17	MB082	100111	ultima_esp	99628/111	100	(CUENTA-J2 CUENTA-JUG2 CAPTURA2 JUGADAS2 DIFERENCIA2 MI-F-EV)
18	MB083	100110	Aporel2	99608/111	100	(CUENTA-J CUENTA-JUG3 CAPTURA3 JUGADAS3 DIFERENCIA3 MI-F-EV)
19	MC042	100112	DL_LZ2	99391/111	98	(MI-F-EV DIF-MAX-MIN POS-PARES-NO-CERO MEJOR-ROBO NO-AMBOS-PARES)
20	TA063	100109	toy-3	98239/111	100	(NOCERO CAPTURA-CEDE DIF-FICHAS MI-F-EV)
21	TA061	100111	toy-1	97691/111	100	(NOCERO CAPTURA-CEDE DIF-FICHAS MI-F-EV)
22	MB032	100112		32303/37	100	(CASILLA-AMENAZADA AMENAZA PROXIMA-GANANCIA MI-F-EV)
23	TD131	100106	Polpri	96206/111	102	(COMBO AMENAZA-A-POSICION NOS-AMENAZAN AMENAZAMOS GET-TABLERO M
24	MC063	100114	Dimitri	32018/37	102	(JUGADA CADENAS ULTIMA-CASILLA BUSCA-PAREJAS MI-F-EV)
25	TD132	100115	Cholo	857	100	(COMBO AMENAZA-A-POSICION NOS-AMENAZAN AMENAZAMOS GET-TABLERO M
26	ME053	100107	4CHAN	94756/111	100	(MI-F-EV M C SQR)
27	MD052	100115	-O_-O-	842	100	(MAYOR CALCULA-PERDIDA CONFIRMA-ROBO CALCULA-ROBO MI-F-EV)
28	ME051	100115	HAMYGABLE	836	60	(MI-F-EV M C SQR)
29	MC113	100113	Scar	92135/111	100	(ROBO-REAL-PAR-IMPARGOBO-REAL-IMPARGOBOSIBILIDADES-DE-ROBO2 MI-F-EV)
30	MC073	100115	REPLICANTE	828	100	(MI-F-EV)
31	TD081	100109		90820/111	100	(ES-ACCESIBLE CUENTA-PAREJAS-ACCESIBLES MI-PAR CUENTA-CEROS DIFEREN
32	MD021	100108	TuttiFruti	90605/111	100	(COMER ANALISIS MI-F-EV)
33	MD091	100115	--	816	100	(NUEVA-WIKI MI-F-EV-MOVILIDAD MI-F-EV-NF MI-F-EV)
34	MD073	100109		20050/37	100	(ROBO2 FICHAS-ROBO2-ROBO2 FICHAS-ROBO2-ROBO2 FICHAS-ROBO2-ROBO2

Figura 1. Clasificación del torneo. Vista del profesor.

## TRABAJO FUTURO

El actual objetivo es desarrollar una plataforma estable que permita a los estudiantes entablar partidas con otros jugadores a través de la red, incorporando una serie de mejoras, incluidas aquellas que los propios alumnos han ido sugiriendo. Esta plataforma permitirá a los estudiantes utilizar los resultados de dichas partidas durante el proceso de diseño y mejora de las estrategias automáticas. De esta forma se pone a disposición de los estudiantes un instrumento de aprendizaje que se espera incremente la motivación de los mismos, al favorecer la participación y la interacción con sus compañeros, y mejore los resultados del aprendizaje, estimulando el desarrollo de estrategias de juego más inteligentes y efectivas.

Esta plataforma, que de modo experimental se puso en funcionamiento durante 4 semanas en el curso 2009/2010, se pretende que sea estable y permanente, con lo que los alumnos podrían utilizarla durante todo el período docente. Asimismo se pretende continuar utilizándola y mejorándola en los sucesivos cursos académicos.

Las mejoras previstas para el próximo curso académico van enfocadas a varios aspectos:

- Estabilizar el sistema ya existente.
- Dotarlo de capacidad para dar soporte a tareas colaborativas, por ejemplo, de la opción de participación en grupo, donde todos los estudiantes jueguen contra un jugador bueno. Los estudiantes podrían ver el código de los otros estudiantes y desarrollar ideas para intentar vencer al jugador bueno (cuya estrategia es desconocida para ellos). Se premiaría a cada estudiante individualmente por las contribuciones hechas y también al colectivo de acuerdo con la clasificación. Ello emularía actividades reconocidamente exitosas en el mundo real como el desarrollo de aplicaciones de código abierto (Linux, proyectos GNU, etc.).
- Dotarlo de capacidad de controlar juegos multiagente.
- Interacción con Moodle. Ya se ha dicho que la funcionalidad implementada excede la que suministra Moodle. No obstante, creemos que sería no sólo factible, sino conveniente enlazar con Moodle, pues ello facilitaría la labor docente, permitiendo combinar pruebas de ambos entornos. Uno de los objetivos de la segunda fase, a implementar durante el curso 2010-2011 es la evaluación del modo en que dicha interfaz sería posible.

## BIBLIOGRAFÍA

1. Dilworth, R. L. (1998). Action Learning in a Nutshell. *Performance Improvement Quarterly*, 11(1)(1998a): 28-43. Disponible en [www.itapintl.com/facultyandresources/articlelibrarymain/](http://www.itapintl.com/facultyandresources/articlelibrarymain/)
2. Dunn, L. (2002). *Theories of learning*. Learning and Teaching Briefing Papers Series. Oxford Centre for Staff and Learning Development OCSLD en [www.brookes.ac.uk/services/ocsd/](http://www.brookes.ac.uk/services/ocsd/)
3. Russell, S., Norvig, P. (1995) *Artificial Intelligence: A Modern Approach*, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey
4. Samuel, A.L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), pp. 210–229
5. Turing, A.M. (1950). Computing machinery and intelligence. *Mind*, 59(236): 433-460.
6. Wittgenstein, L. (trad. 1987). *Tractatus logico-philosophicus*. Madrid, Alianza.

Recibido: 19 marzo 2010.

Aceptado: 31 marzo 2010.