

## Programación-1: Una asignatura orientada a la resolución de problemas

Jordi Petit, Salvador Roura  
Departament de Llenguatges i Sistemes Informàtics  
Universitat Politècnica de Catalunya  
Campus Nord, edifici  $\Omega$   
08034 Barcelona

### Resumen

En este artículo presentamos el nuevo método docente de la asignatura Programació-1 en la Facultat d'Informàtica de Barcelona de la Universitat Politècnica de Catalunya, y damos cuenta de los resultados obtenidos durante los cinco cuatrimestres de aplicación. El curso proporciona un sistema automático de verificación de soluciones para una colección de casi 300 ejercicios cuidadosamente ordenados. Este sistema automático se usa también en la evaluación de los estudiantes durante los exámenes, que se realizan con un ordenador. Además, el sistema permite obtener datos objetivos sobre el progreso de los estudiantes a lo largo del curso.

### 1. Introducción

Programació-1 (Programación-1, P1) es una asignatura del primer cuatrimestre de la Facultat d'Informàtica de Barcelona de la Universitat Politècnica de Catalunya. Esta asignatura es obligatoria para todas sus titulaciones (EI, ETIS y ETIG), cuenta con 7'2 créditos ECTS, y forma parte de la llamada Fase de Selección. En el pasado cuatrimestre, 498 estudiantes se matricularon de P1.

En el curso 2006/2007, P1 se reformó partiendo de la constatación de que muchos estudiantes llegaban a cursos posteriores sin haber conseguido las habilidades mínimas de programación necesarias para implementar fácilmente algoritmos sencillos. Por ello, el objetivo fundamental del actual P1 es que el estudiante sea capaz de escribir con soltura programas correctos y legibles que resuelvan problemas de dificultad elemental.

Para lograr este objetivo, el curso está orientado a la resolución de problemas. La asignatura proporciona una colección de casi 300 ejercicios de pro-

gramación cuidadosamente ordenados y graduados. El método más novedoso y original es *El Jutge de P1* (El Juez de P1), un servicio web para probar programas mediante extensos juegos de pruebas. Con este sistema, los estudiantes pueden diagnosticar en pocos segundos la validez de sus soluciones para cada uno de los ejercicios de la colección. El Jutge de P1 es también parte integrante del sistema de evaluación de la asignatura. Además, el juez proporciona información objetiva sobre el trabajo, aprovechamiento y hábitos de estudio de los estudiantes.

Si bien es cierto que jueces de este tipo han sido ampliamente usados en el campo de los concursos de programación (ver, p.ej. [10, 7]) y nuestro sistema es parecido, creemos que ésta es la primera vez que se utiliza tal herramienta para un curso de introducción a la programación. Algunas experiencias similares pero de menor magnitud y escala pueden encontrarse, por ejemplo, en [1, 3, 9].

La organización del artículo es la siguiente: Primero presentamos brevemente la asignatura. Luego describimos sus principales recursos: la colección de problemas y el juez. A continuación describimos el sistema de evaluación. Finalmente, damos cuenta del impacto de este nuevo enfoque, presentando y analizando los resultados obtenidos durante sus cinco cuatrimestres de aplicación.

### 2. Descripción de la asignatura

#### 2.1. Objetivos y temario

El objetivo fundamental del actual P1 es *que el estudiante sea capaz de escribir con soltura programas correctos y legibles que resuelvan problemas de dificultad elemental*. El temario del curso es muy parecido al de otros cursos de introducción a la programación, si bien casi todo el temario abstracto y de cariz lógico-matemático ha desaparecido (es-

pecificaciones, demostraciones de terminación, demostraciones de corrección, inducción, invariantes, precondiciones, postcondiciones,...). Consideramos que estos conocimientos deben formar parte del repertorio de un ingeniero informático, pero los proponemos para más adelante, y nos centramos en los aspectos más técnicos y prácticos, orientados a lograr la destreza mínima en programación requerida por las siguientes asignaturas de la carrera.

## 2.2. Metodología docente

El curso se organiza en 13 semanas de clases, a razón de 6 horas semanales. Cada semana del curso consiste en una sesión de 3 horas de teoría/problemas y en otra sesión de 3 horas de laboratorio con presencia parcial del profesor. Los grupos de teoría/problemas son de unos 60 alumnos, y los de laboratorio de unos 20, con un ordenador por alumno.

Durante todo el curso, se pretende que el estudiante trabaje por su cuenta, y con apoyo de un profesor, la colección de ejercicios que describimos en la siguiente sección. Asimismo, se espera que envíe sus soluciones al juez automático que comprobará su funcionamiento con juegos de pruebas exhaustivos. Adicionalmente, el estudiante deberá mostrar regularmente sus programas (funcionen o no) a los profesores, para que éstos evalúen su calidad.

## 3. La colección de enunciados

El material más importante para nuestro curso de programación es lógicamente la colección de enunciados. En la actualidad, la colección contiene unos 200 ejercicios [5] y unos 90 problemas de exámenes y controles [4, 2]. En <https://p1.lsi.upc.edu:2001/doc/jenui2009/examenes.pdf> se reproducen en castellano los problemas del último examen final.

La colección de ejercicios está organizada siguiendo el temario. Además, dentro de cada tema, los ejercicios están graduados de acuerdo con su dificultad. Para facilitar la organización del curso, los ejercicios también están ordenados según una programación semanal, correspondiente a las sesiones de laboratorio. Así mismo, los ejercicios se clasifican en tres tipos: los que deberían hacerse durante las sesiones de laboratorio, los que deberían traba-

jarse “en casa”, y otros opcionales que pueden ser más complicados o más largos, pero que pueden resolverse con los conocimientos dados hasta ese momento. No se proporcionan soluciones para ninguno de estos ejercicios.

Los problemas de los exámenes de cursos anteriores y de los controles también forman parte de la colección. Para éstos sí se dan posibles soluciones.

Existen tres clases de enunciados: La mayoría piden la elaboración de un programa completo que, a partir de unas entradas bien definidas, produzca unas salidas específicas. Para estos enunciados, primero se describe el problema en cuestión. A continuación se define con más rigor el formato de las entradas y de las salidas. Finalmente, se dan algunos ejemplos de entradas y sus salidas correctas correspondientes. Así pues, la tarea del estudiante es pensar e implementar un algoritmo completo. La segunda clase de enunciados sólo pide la escritura de una función o acción, y la tercera clase combina las dos primeras.

En cualquier caso, los enunciados siempre describen de forma clara y sin ambigüedades el problema a resolver. En particular, se definen las condiciones que se asumen en las entradas, que nunca deben verificarse con el programa solución.

## 4. El juez

El Jutge de P1 es un entorno virtual de aprendizaje accesible vía web las 24 horas del día para todos los matriculados. Si bien ofrece funcionalidades propias de otros sistemas educacionales (gestión de documentos, tablón de anuncios, publicación de notas, etc), su principal cometido es probar de forma automática la corrección de las soluciones enviadas a los enunciados de la colección.

El uso del juez es como sigue: Primero, el estudiante (o profesor) se identifica al sistema. A continuación, elige un problema de la colección, viendo su enunciado en HTML, PDF o PS, y pudiendo bajarse los juegos de pruebas públicos. Cuando el usuario está convencido de la corrección de su solución, y quiere comprobarla, la envía al juez, el cual le devuelve un veredicto en pocos segundos.

El proceso interno para calcular el veredicto es el siguiente: Primero, el juez compila la solución del estudiante. En caso de error, se devuelve un veredicto de “*Compilación incorrecta*” con los men-

sajes del compilador. En otro caso, el juez intenta ejecutar el programa sobre los juegos de pruebas, tanto públicos como privados. Dichas ejecuciones se realizan en un entorno controlado para evitar la ejecución de instrucciones maliciosas. Si para alguna de estas ejecuciones el programa aborta, se emite un veredicto de “Ejecución incorrecta”, acompañado de la causa del error (división por cero, acceso ilegal a memoria, tiempo de ejecución excedido, ...). Si no, y para cada juego de pruebas, se compara la salida generada con la correcta. Si todas las salidas generadas son idénticas a las correctas, se emite un veredicto de “Supera los juegos de pruebas”<sup>1</sup>. Si no, el juez mira si las diferencias sólo son debidas al formato de la salida (espacios o saltos de línea incorrectos, uso inadecuado de las mayúsculas, ...). En este caso se emite un veredicto de “No supera los juegos de pruebas: error de presentación”. En otro caso, se emite un veredicto de “No supera los juegos de pruebas: respuesta errónea”.

Evidentemente, el sistema de validación de programas requiere de la creación de juegos de pruebas privados para cada enunciado, los cuales deben comprobar la corrección de las soluciones recibidas de forma tan exhaustiva como sea posible, así como la eficiencia cuando ésta sea uno de los requerimientos del enunciado en cuestión. A menudo, cuesta mucho más crear unos buenos juegos de pruebas que pensar, escribir y solucionar los enunciados.

Por otro lado, cuando un programa supera todos los juegos de pruebas, es muy posible que “funcione”, pero no se puede garantizar que sea “correcto”: puede contener errores imprevisibles que no se han detectado en los juegos de pruebas, puede utilizar un algoritmo claramente inadecuado, o puede estar muy mal escrito. Por ello, durante el curso se recomienda encarecidamente que los estudiantes muestren regularmente a los profesores sus programas aceptados por el juez. A pesar de esta limitación, el valor del juez es evidente en un curso de introducción a la programación, donde los estudiantes escriben programas erróneos con tanta frecuencia.

En cuanto a los profesores de la asignatura, el

<sup>1</sup>De hecho, el juez dispone de sistemas de corrección más sofisticados. Por ejemplo, puede proporcionar pistas sobre porqué determinados juegos de pruebas no son superados. Además, y aunque no se use en P1, en determinados casos se podría permitir que el orden de la salida sea arbitrario, o establecer un límite en la precisión de los resultados.

juez también les ofrece interesantes funcionalidades. En particular, los profesores pueden monitorizar a un estudiante, viendo todos sus envíos, diagnosticar rápidamente sus errores gracias a la visualización gráfica de las diferencias entre la salida generada por su programa y la correcta, y comunicar sus consejos vía correo electrónico. Además, el juez ofrece estadísticas en tiempo real sobre el progreso de todos los estudiantes matriculados, con la posibilidad de desglosar los resultados por grupos.

## 5. Sistema de evaluación

El sistema de evaluación de la asignatura se ha diseñado de forma coherente con el objetivo principal del curso y el método docente utilizado. Por ello, los exámenes siguen el mismo patrón y dificultad media que los enunciados de la colección, y se realizan delante de un ordenador y usando el juez.

Concretamente, hay varias pruebas, todas consistentes en resolver un problema. Durante el curso, y para potenciar la evaluación continuada, se realizan 6 controles que evalúan el seguimiento del temario presentado hasta el momento. Los controles son cada 2 o 3 semanas, y en horas de clase.  $C$  denota la media ponderada de los controles. En el periodo de exámenes, se realizan tres pruebas ( $F_1$ ,  $F_2$  y  $F_3$ ) que evalúan la totalidad del temario.

Cada problema se puntúa independientemente de los otros y recibe una nota entre 0 y 10 puntos. La nota final de la asignatura<sup>2</sup> es  $\max\{(C + F_1 + F_2 + F_3)/4, (F_1 + F_2 + F_3)/3\}$ .

Durante los exámenes, cuando un estudiante tiene un programa que considera correcto, lo envía al juez, el cual le devuelve su veredicto. Tanto si el programa pasa todos los juegos de pruebas como si no, el estudiante puede reenviar varias soluciones para el mismo problema. Si al final del examen no ha enviado ningún programa que supere todos los juegos de pruebas, su nota es 0.

De lo contrario, los profesores corrigen manualmente el último programa que supera los juegos de pruebas. Si se detecta que el estudiante se ha saltado las reglas básicas de programación fijadas con claridad durante el curso [8], o si el programa no cumple los requerimientos del enunciado, o si el algoritmo

<sup>2</sup>Ésta es la fórmula actual. Durante los tres primeros cuatrimestres, el sistema de evaluación no incluía controles, pero sí un parcial recuperable.

usado para resolver el problema es manifiestamente inadecuado, la nota es 0.

De lo contrario, sea  $V$  la valoración, entre 0 y 5, dada por los profesores al programa corregido (en función de la claridad, el estilo, etcétera), y sea  $P$  el número de programas enviados que no superan los juegos de pruebas. La nota es  $5 + V - \max\{P - 1, 0\}$ . Es decir, se resta un punto sobre diez por cada solución enviada que no pase los juegos de pruebas, si bien la primera solución errónea no se penaliza. Lógicamente, una nota negativa se convierte en 0 (hecho que acota a 10 el máximo número de envíos incorrectos).

El tiempo para resolver un problema de examen es de una hora. Además, al principio del examen los estudiantes reciben 15 minutos adicionales durante los cuales pueden leer el enunciado y escribir en papel, pero no usar el ordenador (hemos constatado que este tiempo no sólo obliga a leer y a reflexionar, sino que sirve para reducir los nervios). También, los estudiantes reciben 5 minutos adicionales al finalizar la hora, para poder entregar soluciones fuera de tiempo con una penalización proporcional al retraso.

Por otra parte, durante los exámenes cada estudiante puede consultar una “chuleta” electrónica (1000 caracteres máximo) que ha debido registrar en el sistema con una semana de antelación.

## 6. Evaluación del impacto

En esta sección analizamos los resultados de la aplicación de nuestro método desde diversos puntos de vista y, en particular, usando los datos obtenidos gracias al juez.

Los datos referidos son los del curso 2008/2009 Q1 con 498 matriculados (368 nuevos, 130 repetidores)<sup>3</sup>; para los otros cuatro cursos, los resultados son perfectamente semejantes. Al final de esta sección damos unos datos generales sobre los cinco cuatrimestres.

### 6.1. Trabajo durante el curso

P1 tiene 7'2 ECTS. Cada ECTS se corresponde a entre 25 y 30 horas de trabajo de un estudiante me-

<sup>3</sup>498 es el número de matriculados el día del examen final. Las tablas y gráficos muestran datos para más matriculados, a causa de los cambios en los listados.

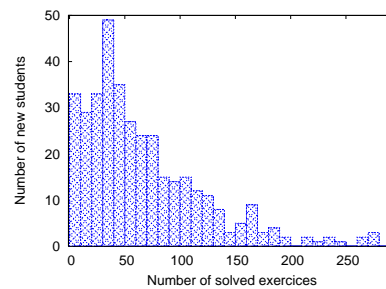


Figura 1: Trabajo durante el curso (1)

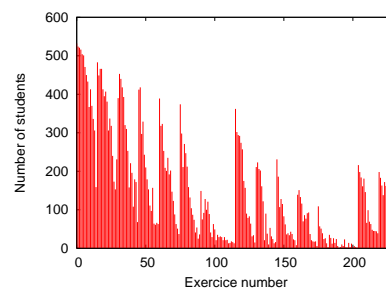


Figura 2: Trabajo durante el curso (2)

dio. Multiplicando, y restando 45 horas de clases de teoría (que de hecho son menos), nos quedan entre 135 y 170 horas de práctica. Supongamos ahora que resolver un ejercicio requiere en promedio 45 minutos (evidentemente esto depende mucho del tema y de la dificultad, pero resolver 4 ejercicios en una sesión de laboratorio de 3 horas es de sobras factible). Entonces, un estudiante debería resolver entre 180 y 225 ejercicios. En todo caso, quedémonos con la cantidad inferior de 180. Y aún en el caso que, en media, el tiempo requerido para resolver un ejercicio fuese de una hora, entonces ese número debería ser 135, como mínimo.

El histograma de la figura 6 muestra la distribución del número de problemas resueltos por estudiante. Contando los 6 controles y los 3 problemas de examen, el número de ejercicios resueltos en media por cada estudiante nuevo de P1 es 68. Para estudiantes repetidores, este número es 85. La mediana es 58. Sólo 66 estudiantes han resuelto más de 135 ejercicios. Sólo 30 estudiantes han resuelto más de 180 ejercicios.

Por otro lado, el histograma de la figura 2 mues-

$n$ : nota	C1	C2	C3	C4	C5	C6
$n \geq 9$	199	139	106	96	137	46
$5 \leq n < 9$	133	74	52	45	46	28
$0 < n < 5$	5	6	6	4	4	3
$n = 0$	105	207	246	277	143	149
$n = \text{NP}$	56	72	88	70	168	271

Cuadro 1: Distribución de notas en controles

tra cuantos estudiantes no repetidores han resuelto cada ejercicio de la colección, ordenando los ejercicios según aparecen en la colección. La animación <https://p1.lsi.upc.edu:2001/doc/jenui2009/animacion.gif> muestra cómo evoluciona el número de problemas resueltos (por los no repetidores) a medida que el curso avanza, con granularidad diaria. En verde se muestra el número de estudiantes que resuelve correctamente cada ejercicio y, en rojo, el número de estudiantes que envía una solución para ese ejercicio pero no consigue superar los juegos de pruebas.

Como se puede apreciar, el número de ejercicios realizados baja según avanza el curso, con unos rebotes al empezar cada nueva sesión de laboratorio (y un gran rebote al empezar la segunda parte del curso, tablas). Además, la animación temporal constata el retraso de muchos estudiantes a medida que avanza el curso. En cualquier caso, es evidente que la inmensa mayoría de los estudiantes que envía soluciones para un problema llega a resolverlos.

En definitiva, el número de ejercicios resueltos durante el curso es, globalmente, extremadamente bajo.

## 6.2. Resultados de la evaluación continuada

El cuadro 1 muestra la distribución de las notas durante los 6 controles del cuatrimestre. Como se puede observar, el número de No Presentados se incrementa paulatinamente a medida que avanza el curso, mientras que el número de aprobados desciende. Casi todos los suspensos son causados por no haber enviado ninguna solución que supere los juegos de pruebas.

Teniendo en cuenta la sencillez de los controles (excepto el C4, correspondiente al parcial), queda claro que la mayoría de los estudiantes no siguen el ritmo del curso. Además, es sorprendente ver cómo desde la tercera semana (fecha de realización del primer control) un 10% de los estudiantes re-

nuncia a la evaluación continuada, y un 20% de los estudiantes no son capaces de hacer un programa sin bucles ni procedimientos para un enunciado del tipo “leer cinco enteros  $x, a, b, c, d$  y decir si  $x \in [a, b] \cup [c, d]$ ”.

## 6.3. Preparación de los exámenes

Los datos sobre la preparación de los exámenes finales son muy ilustrativos. La última sesión del curso contiene 20 ejercicios de consolidación, de un nivel de dificultad parecida o superior a los exámenes finales, para poderlos preparar a conciencia. En media, cada estudiante resolvió 1'02 de estos ejercicios. Sólo 128 de los 498 estudiantes resolvieron alguno. La colección de ejercicios también incluye los 20 enunciados de los exámenes finales de P1 de los cuatro cursos anteriores. En media, cada estudiante resolvió 2'44. Sólo 156 de los 498 estudiantes resolvieron alguno.

Recordemos también que los estudiantes pueden registrar una “chuleta electrónica” que pueden consultar durante los exámenes. Sólo un 56% de los estudiantes preparó su chuleta para el parcial y sólo un 40% lo hizo para el final.

Finalmente, en la animación referida anteriormente también puede observarse una caída de rendimiento durante el periodo de Navidad, hecho bastante sorprendente puesto que el curso acabó el 23 de diciembre y el examen final era el 7 de enero.

Todos estos hechos muestran una escasa preparación para los exámenes.

## 6.4. Resultados de la evaluación final

La figura 3 muestra la distribución de las notas finales. El número MH es 14, el de otros aprobados 89, el de “compensables” (nota entre 4 y 5) 29, el de otros suspensos 201, de los cuales 67 son cero. En la gráfica no se muestran los 164 estudiantes no presentados.

Se puede observar que las notas no siguen una distribución normal, sino que muestran una concentración alrededor del cero, el tres/cuatro, seis y diez. Esto es consecuencia de la fórmula de evaluación, la cual permite discriminar claramente las capacidades demostradas por los estudiantes, información vital en una asignatura de Fase de Selección.

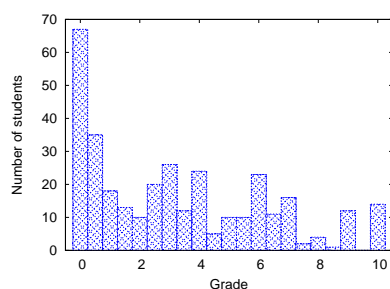


Figura 3: Notas finales

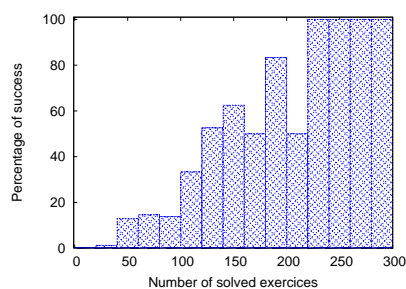


Figura 4: Correlación problemas/aprobados

### 6.5. Relación trabajo-resultado

A la luz de los datos anteriores, es lógico preguntarse si el esfuerzo de los estudiantes influye en su nota. Evidentemente la casuística es muy grande, pero los datos son significativos: El número medio de ejercicios resueltos por estudiante nuevo ha sido de 211 para los estudiantes con MH; 120 para los otros aprobados; 96 para los estudiantes con “compensable”; 71 para los otros suspensos; y 27 para los no presentados.

De forma más general, la figura 4 muestra la correlación entre el número de problemas realizados y la tasa de aprobados. Como es de esperar, la tendencia es que las posibilidades de aprobar se incrementan a medida que aumenta el número de problemas resueltos.

### 6.6. Historial de resultados

El cuadro 2 contextualiza los porcentajes de aprobados de P1 con los de las demás asignaturas de primer cuatrimestre. Presentamos los porcentajes sobre el número de presentados por razones de espa-

cio y para hacer comparables los resultados desde el punto de vista de la dificultad de los exámenes, pero los resultados con NPs son equivalentes. Además, los aprobados de P1 no contabilizan los estudiantes convalidados provenientes de Módulos Formativos.

Como puede constatar, los resultados de P1 son siempre parecidos a los de Álgebra (con la excepción del último cuatrimestre, en el cual esta asignatura cambió su sistema de evaluación), siendo ambos claramente inferiores a los de Introducción a los Computadores y con los resultados de Física entre ellos. Por razones de espacio, no analizamos aquí estas diferencias; el lector interesado puede consultar [6].

### 6.7. Críticas recibidas

El método docente de P1 ha recibido numerosas alabanzas por parte de estudiantes, profesores, decanato y rectorado. Sin embargo, todos estos estamentos —con la excepción de los profesores que han impartido la asignatura— han criticado su sistema de evaluación, no tanto por el sistema en sí, sino por el número de aprobados resultante. En efecto, si bien P1 no es determinante, sí juega un papel importante en el impacto de la Fase de Selección y, en consecuencia, de toda la carrera.

Como responsables de la asignatura, hemos recibido presiones y hemos acudido a numerosas entrevistas para justificar estos resultados. En estos casos, nuestra tarea ha sido siempre exponer que los estudiantes que trabajan el curso en general lo aprueban, y que los estudiantes que no trabajan en general suspenden. Aquí, entendemos que trabajar el curso significa probar de resolver sus ejercicios. Además, los resultados recopilados por el juez durante el curso nos permiten demostrar a nuestros interlocutores que la mayoría de matriculados no estudia. Evidentemente, esto abre interrogantes en relación a la motivación de nuestros estudiantes.

Por otra parte, se nos ha criticado con frecuencia el gran número de ceros que aparecen en las evaluaciones finales. Evidentemente, estos ceros no significan forzosamente que no se haya aprendido nada durante el curso, pero sí significan que, de nueve problemas asignados durante el curso, no se ha sabido realizar ningún programa correcto para ninguno de ellos. Entendemos que ello es una prueba contundente de su inmadurez para pasar el curso.

Curso	P1	AL	F	IC
2006/2007 1Q	31	31	47	56
2006/2007 2Q	33	36	35	56
2007/2008 1Q	32	29	47	64
2007/2008 2Q	49	48	41	66
2008/2009 1Q	31	52	42	57

Cuadro 2: Porcentajes aprobados de las asignaturas de 1Q

## 7. Conclusiones

Hemos presentado el nuevo método docente de P1, puesto en marcha desde hace cinco cuatrimestres. La particularidad principal de este método es su enfoque extremadamente práctico orientado a la resolución de problemas. Los pilares de esta aproximación son, por un lado, una extensa colección de problemas cuidadosamente seleccionados y ordenados y, por otro lado, un juez automático que permite validar en tiempo real los programas escritos por los estudiantes.

A nuestro parecer, el uso del juez tiene múltiples ventajas:

- Como herramienta de aprendizaje, es el mecanismo ideal para aprender a programar, ya que contiene un extenso listado de ejercicios, evalúa las soluciones enviadas por los alumnos a lo largo del curso y les informa inmediatamente sobre si son o no correctas (según juegos de pruebas muy exhaustivos).
- Como herramienta de ayuda a la evaluación, el juez permite desarrollar exámenes delante del ordenador, donde el alumno es evaluado precisamente de aquello que debe aprender, es decir, resolver problemas de programación sencillos (saber leer el enunciado de un problema, pensar un algoritmo para el mismo, implementarlo, y comprobar que funcione). Además, el juez es objetivo e imparcial.
- Como herramienta de seguimiento del progreso de los estudiantes, el juez proporciona una fuente de datos de una objetividad desconocida hasta el momento. En efecto, el juez es parecido a un "Big Brother" que sigue los avances de los estudiantes, problema a problema, tanto con estadísticas globales sobre todos los

matriculados, como de datos restringidos a un grupo o a un estudiante en particular.

En relación a los resultados, hemos demostrado una gran correlación entre el trabajo de los estudiantes y las notas que obtienen. Lamentablemente, los datos recojidos con el juez confirman la sospecha de que el esfuerzo de la mayoría de nuestros estudiantes dista mucho de la carga prevista dentro del marco de créditos ECTS. Como consecuencia, P1 tiene un número elevado de suspensos.

El hecho de que la única crítica que reciba P1 sea sobre su sistema de evaluación, y con el triste argumento de un número de suspensos "muy alto", nos confirma que P1 está impartida por buenos profesores, está bien organizada, dispone de un excelente material y exhibe un enfoque ligero y moderno. En definitiva, estamos convencidos que P1 es un perfecto exponente de la adaptación al Espacio Europeo de Educación Superior, que propugna asignaturas de carácter práctico que incorporen nuevas metodologías docentes dando énfasis en el aprendizaje basado en proyectos y problemas.

## Referencias

- [1] Douce C., Livingstone D., and Orwell J. Automatic test-based assessment of programming: A review. *ACM Journal on Educational Resources in Computing*, 5(3), 2005.
- [2] Duch A., Petit J., and Roura S. Col·lecció de controls de P1 solucionats, 2008. <https://p1.lsi.upc.edu:2001/pbm/controls.pdf>.
- [3] Joy M., Griffiths N., and Boyatt R. The boss online submission and assessment system. *ACM Journal on Educational Resources in Computing*, 5(3), 2005.
- [4] Petit J. and Roura S. Col·lecció de problemes d'exàmens de P1 solucionats, 2009. <https://p1.lsi.upc.edu:2001/pbm/examens.pdf>.
- [5] Petit J. and Roura S. Col·lecció d'exercicis de P1, 2008. <https://p1.lsi.upc.edu:2001/pbm/exercicis.pdf>.
- [6] Petit J. and Roura S. Resposta a la Carta al respecte de l'avaluació de l'assignatura P1 a la fib, 2009. <https://p1.lsi.upc.edu:2001/doc/resposta-dafib.pdf>.

- [7] Revilla M. Uva online judge.  
<http://icpcres.ecs.baylor.edu/onlinejudge/>.
- [8] Roura S. Normes de programació de P1, 2008.  
<https://p1.lsi.upc.edu:2001/doc/normes.pdf>.
- [9] Saikkonen R., Malmi L., and Korhonen A. Fully automatic assessment of programming exercises. In *Proceedings of ITICSE' 2001, Innovation and Technology in Computer Science Education*, pages 133–136, 2001.
- [10] Skiena S. and Revilla M. *Programming Challenges*. Springer, 2003.