

Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programación - Experiencia en la Universidad Nacional de Colombia

Julian Moreno
Universidad Nacional de Colombia
Cra 80, 65-223, Of M8A 311
Medellín, Colombia
(+574)425221
jmoreno1@unal.edu.co

Andrés F. Pineda
Universidad Nacional de Colombia
Cll 61 #53-80, Apt 302
Medellín, Colombia
(+57) 2318982
afpinedac@unal.edu.co

Luis F. Montoya
Universidad Nacional de Colombia
Cra 28 # 30-08, Apt 302
Marinilla, Colombia
(+57) 5690689
lfmontoyag@unal.edu.co

RESUMEN

En este artículo se describe la experiencia dentro de la Universidad Nacional de Colombia, más específicamente con estudiantes de pregrado de la Sede Medellín, del uso de PPC (*Programming Practice Center*). PPC es una plataforma virtual que permite a los profesores crear y administrar cursos relacionados con la programación de computadores a partir de ejercicios de tipo ACM-ICPC. La característica principal de dicha plataforma es que emplea un juez en línea, donde los estudiantes pueden mantener un registro de su avance y desarrollar sus habilidades en diferentes temas mediante la resolución de problemas al tiempo que compiten con sus compañeros.

ABSTRACT

This article describes the experience in the National University of Colombia, more specifically with undergraduate students, using PPC (*Programming Practice Center*). PPC is a virtual platform that enables teachers to create and manage courses related to computer programming exercises from ACM-ICPC type. The main feature of this platform is that it employs an online judge where students can keep track of their progress and develop their skills in different subjects by solving problems while competing with peers.

Categorías

F.2 [Análisis de algoritmos y complejidad de problemas], K.3 [Computadores y educación]

Términos generales

Algoritmos.

Palabras clave

Ciencias de la computación, PPC, Programación competitiva, ACM-ICPC

1. INTRODUCCIÓN

Usualmente dentro de los pregrados relacionados con las ciencias de la computación, los profesores deben estructurar cursos de programación en los que hay una gran cantidad de estudiantes. Esta situación dificulta en cierto grado un tipo de enseñanza personalizada, en donde se pueda asegurar que el estudiante ha entendido y adquirido los conceptos que el profesor le ha querido transmitir [4].

Este tipo de cursos sobrepoblados normalmente son abordados por los profesores con clases magistrales y exámenes, donde solo se dictan las clases mientras que los estudiantes escuchan y en los que solo hay una interacción profesor-estudiante. El problema general de esta metodología es que empieza a hacer que los estudiantes se aburran lo cual conlleva a que con el paso del tiempo se vaya perdiendo poco a poco el interés en los temas abordados en el curso.

Varios autores [5, 11, 7] proponen varias metodologías para hacer que este tipo de interacción se lleve más adecuadamente y se pueda incrementar la motivación y el compromiso de los estudiantes en un ambiente educativo. Todos ellos concuerdan en el hecho de que hacer uso de aplicaciones para e-learning es un método efectivo de enseñanza que permite romper las barreras en los cursos.

Otro factor importante a la hora de obtener más compromiso y acogida de un curso por parte de los estudiantes son los incentivos [8, 2]. Esto bajo el supuesto que el ser humano se siente más motivado cuando su esfuerzo es recompensado y notado de alguna manera por los demás.

Por lo anteriormente es mencionado es que en los últimos años se ha venido desarrollando alrededor del mundo la 'Programación Competitiva'. Esta es una metodología que promueve que los estudiantes escriban programas para resolver una serie de problemas propuestos y posteriormente tales soluciones son probadas con unos casos de prueba que validan la correcta implementación de los algoritmos involucrados.

Considerando el panorama previamente descrito, el resto del artículo está organizado como sigue. En la sección 2 se ahonda un poco más sobre el mundo de la programación competitiva. Luego, en la sección 3, se describe una plataforma llamada PPC y desarrollada al interior de la Universidad Nacional de Colombia para promover precisamente esa práctica. Posteriormente en la sección 4 se presenta un estudio de caso dentro de esa institución para luego, en la sección 5, presentar los resultados y un breve análisis. Finalmente en la sección 6 se discuten las conclusiones y el trabajo futuro.

2. EXPERIENCIAS CON LA PROGRAMACIÓN COMPETITIVA

LA ACM (Association for Computing Machinery) anualmente organiza La Competición Internacional Universitaria de Programación ICPC (International Collegiate Programming

Contest), una competencia de programación y algorítmica entre universidades de todo el mundo patrocinada por IBM.

Diversas universidades han implementado cursos de algoritmos y algoritmos basados en ejercicios tipo ACM-ICPC [3, 9, 6], los cuales promueven la competitividad. Para esto hacen uso de diferentes plataformas [1, 8] obteniendo buenos resultados en términos de rendimientos académicos.

Diferentes autores han destacado las ventajas que puede tener el adoptar un sistema de estos como apoyo dentro de un curso de programación [3, 4]. Por ejemplo por parte del profesor, este realiza una calificación automática lo cual le permite ahorrar tiempo y poder invertirlo en otras áreas de enseñanza y por parte del estudiante le permite tener su calificación inmediatamente, mantener estadísticas de su progreso a lo largo del curso, recibir realimentación inmediata lo cual le permite trabajar tanto dentro como fuera de clase, además se requiere que el estudiante sea mucho más riguroso al momento de realizar un programa.

3. DESCRIPCIÓN DE LA PLATAFORMA PPC

PPC (*Programming Practice Center*) es una plataforma web que se desarrolló dentro de la Universidad Nacional de Colombia con el fin de apoyar a profesores y monitores en los cursos de programación. Más específicamente, dicha plataforma permite definir una serie de problemas para que los estudiantes resuelvan a lo largo del semestre. Para lograr esto, PPC hace uso de la API del juez en línea *UVa Judge Online* de la Universidad de Singapur¹, la cual permite acceder a estadísticas en tiempo real de los usuarios y los ejercicios que estos han enviado al juez. Tales ejercicios son organizados en la plataforma dentro de una serie de temas, los cuales pueden o no tener como pre-requisitos a otros. Esta organización se traduce en una representación gráfica mediante un grafo dirigido donde los nodos corresponden a los temas que se desean abordar y las aristas a las relaciones o pre-requisitos entre dichos temas.

PPC tiene dos tipos de vistas, la vista profesor y la vista estudiante, las cuales se describen a continuación.

3.1 Vista profesor

La plataforma permite que el profesor cree temas, los enlace con otros por medio de relaciones de pre-requisitos y les agregue ejercicios. Adicionalmente le permite establecer para cada tema el número mínimo de ejercicios que el estudiante debe resolver para considerarse que ya ha sido desarrollado satisfactoriamente. Cuando esto sucede dicho tema le aparecerá “desbloqueado” al estudiante, permitiéndole continuar con sus post-requisitos.

En la figura 1 se muestra un ejemplo de este proceso. En este caso se muestra como dentro de un curso el tema “Backtracking”, que contiene 5 ejercicios, es pre-requisito del tema “Programación dinámica” que contiene 8. También se muestra que la cantidad mínima de ejercicios para aprobar “Backtracking” es de 3, cantidad a partir de la cual dicho tema se desbloquearía habilitando así “Programación dinámica” que a su vez tiene una cantidad mínima de 5 ejercicios para aprobarlo.



Figura 1. Ejemplo de vista del profesor durante la creación de un curso

Adicional al panel de creación y modificación de cursos, el profesor cuenta con un panel de monitoreo donde se encuentra información tanto resumida como detallada respecto a cada curso y a los estudiantes inscritos dentro de ellos. Desde este panel un profesor tiene acceso a datos como la cantidad de ejercicios resueltos por un determinado estudiante, los envíos que ha realizado junto con su hora de envío, los correspondientes temas que ha desbloqueado, entre otros. La figura 2 muestra un ejemplo de este panel de monitoreo.

Tema	# de ejercicios resueltos	Mínimo
Introducción	6/7	4
Gráficos - Generalidades	0/3	1
Gráficos - Dijkstra	0/2	1
Gráficos - DFS	0/3	1

Ejercicio	Fecha	Resultado
Palindromes	19-08-2013 03:49:17	Runtime error
Palindromes	19-08-2013 23:08:02	Wrong answer
Palindromes	19-08-2013 23:46:54	Accepted
Word Scramble	19-08-2013 22:50:13	PresentationE
Word Scramble	19-08-2013 22:56:37	Accepted
The Department of Redundancy Department	24-08-2013 01:49:00	Wrong answer
The Department of Redundancy Department	28-08-2013 01:26:40	Wrong answer
The Department of Redundancy Department	28-08-2013 01:31:19	Accepted

Figura 2. Ejemplo de vista del profesor durante el monitoreo de un curso

3.2 Vista Estudiante

Cuando un estudiante se matricula en un curso, previa creación de una cuenta de usuario en *UVa Judge Online*, este podrá ver su grafo del curso similar a como se muestra en la figura 1. Cada vez que un estudiante resuelve satisfactoriamente un ejercicio, su grafo se actualizará automáticamente subrayando en verde dicho ejercicio. Si adicionalmente, con la solución de ese ejercicio el estudiante completa la cantidad mínima aprobatoria del tema correspondiente, este le aparecerá desbloqueado, habilitándole el acceso a los temas que tenga como post-requisitos.

¹ <http://uhunt.felix-halim.net/api>

4. ESTUDIO DE CASO

Como se mencionó en la sección anterior la plataforma PPC se desarrolló dentro de la Universidad Nacional de Colombia con el fin de apoyar a profesores y monitores en los cursos de programación. Más específicamente, su “nacimiento” se dio a partir del curso ‘Semillero de programación’ el cual se ha dictado desde el semestre 2012-1 y al cual asisten generalmente estudiantes del programa ‘Ingeniería de sistemas e informática’.

En la actualidad se encuentran alrededor de 100 estudiantes activos en la plataforma y, adicional a ‘Semillero de programación’, en el semestre 2013-2 se sumó el curso ‘Análisis y diseño de algoritmos’. Como este último curso aún no ha culminado siquiera un período académico, el estudio de caso presentado en esta sección se limita al primero.

Al realizar el diseño curricular y pedagógico de dicho curso, y teniendo en cuenta el soporte en PPC, se tuvieron en cuenta cuatro principios fundamentales que según varios autores [3, 4, 11] deberían estar presentes en todo curso de programación basado en e-learning. Tales principios son:

- **Motivación:** Considerando las altas tasas de deserción de muchos cursos de pregrado, incrementar la motivación de los estudiantes es claramente esencial. Por motivación se quiere decir el deseo a aprender nuevos conceptos y métodos, y ponerlos en práctica.
- **Aprendizaje activo:** Los estudiantes tienen que estar involucrados y conscientes de sus propios procesos de aprendizaje. Una metodología activa donde los estudiantes no son solo espectadores es necesaria para lograr un aprendizaje de largo plazo.
- **Trabajo independiente:** El mejor método para aprender programación de computadores es programando. La simple memorización de conceptos es prácticamente inútil. Los estudiantes tienen que reflexionar sobre los problemas algorítmicos por ellos mismos y crear sus propios programas.
- **Retroalimentación del proceso de aprendizaje:** Tradicionalmente los estudiantes solo obtienen una calificación en ciertos momentos del periodo académico o incluso sólo al final del curso. Contrario a esta práctica, debería existir una evaluación y realimentación continua del progreso de cada uno de ellos.

A partir de estos principios se decidió que la mejor manera para llevar a cabo ‘Semillero de Programación’ sería dividiendo el curso en tres partes:

- 1) **Clases teóricas:** En estas sesiones presenciales el profesor expone los conceptos, demostraciones y algoritmos necesarios que son claves para ayudarle a entender un determinado tema a los estudiantes y que le servirán como base para poder solucionar o afrontar un problema de una manera eficiente.
- 2) **Prácticas permanentes:** El estudiante será el encargado de poner a prueba los conocimientos adquiridos tanto durante las clases magistrales, como de su estudio personal y de la interacción con sus demás compañeros.
- 3) **Competencias:** Después que se han cubierto visto cierta cantidad de nodos (temas) durante el curso, el profesor es el

encargado de organizar competencias de programación entre los estudiantes, donde los ejercicios serán acerca de los temas que se han visto previamente. Allí el estudiante tendrá un mecanismo adicional a los provistos por PPC para demostrar cuanto ha podido avanzar en su estudio personal y medirse para ver cómo está en relación con sus demás compañeros.

Esta estrategia de enseñanza – practica – competencia tiene varios objetivos. Uno de ellos y quizás el más importante es generar lazos tanto de amistad como de cooperación y de competencia entre los estudiantes. Esto con el fin de proporcionarles una alta habilidad de resolución de problemas, un alto nivel de preparación en programación, e incluso “entrenarlos” para la participación en diversos eventos como el Google Code Jam², el Facebook Hacker Cup³, entre otras [10].

5. RESULTADOS Y ANÁLISIS

Con el fin de determinar el nivel de aceptación de la plataforma por parte de la población de estudio anteriormente descrita se decidió realizar una encuesta durante el semestre 2013-2:

Pregunta 1: ¿Cuánto cree usted que ha mejorado sus habilidades en los diferentes temas expuestos en el curso?

Pregunta 2: ¿Cuán motivado se siente usted para superar el número de ejercicios resueltos por sus compañeros?

Pregunta 3: ¿Cómo calificaría de forma general la plataforma?

Las dos siguientes preguntas eran binarias, con posibilidades de respuesta SI/NO:

Pregunta 4: Haciendo uso de la plataforma ¿Se siente motivado a solucionar más problemas?

Pregunta 5: ¿Cree que la plataforma le ha ayudado a estructurar su estudio?

Finalmente, la sexta pregunta era de respuesta libre:

Pregunta 6: Si tiene algún comentario sobre la plataforma, por favor escríbalo a continuación.

En total, solo 13 de los cerca de 50 estudiantes que participan en el curso respondieron la encuesta hasta el día 23 de Agosto de 2013. Cabe mencionar que la misma se implementó mediante Google forms y las respuestas se encuentran disponibles en: <http://goo.gl/BjcnmM>.

La tabla 2 muestra el resumen de los resultados para las tres primeras preguntas.

Tabla 2. Resultados de las preguntas 1-3

Pregunta	Promedio	Desviación estándar
1	7,65	2,14
2	7,25	2,34
3	8,92	0,76

Tanto en la pregunta 1 como en la 2, pese a que las desviaciones estándares son significativas, puede observarse que los estudiantes

² <https://code.google.com/codejam/>

³ <https://www.facebook.com/hackercup>

dieron una alta calificación en ambas. Es decir, por una parte opinan que sus habilidades han mejorado con el uso de la plataforma, y por otra que se sienten motivado para realizar más ejercicios tratando de superar a sus compañeros de curso y así subir posiciones dentro del ranking. Esta segunda pregunta está relacionada con el papel de la competencia entre compañeros y es la que exhibe un menor promedio y una mayor desviación estándar. Esto se puede deber en parte a que algunos estudiantes no se sienten con la suficiente experiencia en la resolución de este tipo de problemas lo que puede llevarlos a pensar que posiblemente no logren alcanzar a algunos de sus compañeros.

Con respecto a la pregunta 3 se puede observar que es la que exhibe un mayor promedio, cercano a 9 puntos sobre 10 posibles, y una muy baja desviación estándar. Esto quiere decir que la impresión generalizada de los estudiantes, incluso de aquellos que sienten que no tienen suficiente experiencia, es que la plataforma les es realmente útil.

En la pregunta 4, que se refiere al aspecto de trabajo independiente y auto-superación, el 100% de los estudiantes respondió afirmativamente. En otras palabras, todos ellos sintieron que la plataforma los motivaba a solucionar cada vez más ejercicios.

Respecto a la pregunta 5, la gran mayoría de los estudiantes (el 92%) respondió que la plataforma les ha ayudado a orientar su estudio personal en los diferentes temas del curso.

Finalmente la pregunta 6 tuvo muy poca participación, sin embargo, un punto común entre las respuestas recibidas es que sería beneficioso complementar la plataforma incluyendo un módulo donde los estudiantes puedan intercambiar códigos de un determinado ejercicio pero al que solo puedan acceder aquellos que ya lo hayan solucionado de modo que puedan comparar aproximaciones.

6. CONCLUSIONES Y TRABAJO FUTURO

En concordancia con los hallazgos de varios autores citados a lo largo de este artículo nuestra opinión luego del uso de la plataforma PPC es que la utilización de un ambiente virtual competitivo dentro de un salón de clase en cursos relacionados a la programación puede generar un alto nivel de motivación por parte de los estudiantes lo cual generalmente se traduce en un mejor desempeño académico.

Nuestra hipótesis es que parte de este efecto positivo se debe a que el uso de la programación competitiva como forma de evaluación resulta para los estudiantes más interesante que los exámenes tradicionales en donde por lo general se estudia mucho contenido en muy poco tiempo (por lo general solo los días antes del examen). En oposición a esta situación, la programación competitiva promueve el estudio constante a lo largo del curso, con la ventaja adicional que les permite a los estudiantes obtener una respuesta inmediata impulsando así el auto mejoramiento.

Y las ventajas no solo son para los estudiantes. A los profesores les permiten tener un monitoreo total sobre el avance y el tiempo de estudio de sus estudiantes, al tiempo que les brinda un sistema de calificación completamente objetiva.

Como trabajo futuro, y desde el punto de vista de la funcionalidad de la plataforma PPC, se plantea la implementación de los módulos necesarios para poder tener acceso a las APIs de otros jueces en línea online como TopCoder o CodeForces. De esta manera se podría ampliar substancialmente el abanico de ejercicios disponibles.

Otro modulo que se planea agregar tiene que ver con las respuestas libres obtenidas mediante la encuesta realizada. Esto es, uno que permita el intercambio de códigos entre los estudiantes que han realizado un determinado ejercicio.

Por último se plantea la posibilidad de incorporar en la plataforma características de ludificación con el fin de explotar aún más el aspecto de la motivación. Entre otras se piensa implementar sistemas de recompensa de tipo medallas o trofeos que se irían desbloqueando a medida que se van resolviendo problemas durante un curso.

AGRADECIMIENTOS

Los autores agradecemos a Sebastián Múnera y Alejandro Escobar, egresados de la Universidad Nacional de Colombia – Sede Medellín, por haber comenzado con el semillero de programación. Igualmente, agradecemos a *UVa hunting* por publicar la API que permitió el desarrollo de PPC, y por dar el soporte necesario.

REFERENCIAS

- [1] Enström, E., Kreitz, G., Niemelä, F., Söderman, P. and Kann, V. Five Years with Kattis – Using an Automated Assessment System in Teaching. 41st ASEE/IEEE Frontiers in Education Conference, USA
- [2] Fraja, G. and Landeras, P. 2005. Could do better: The effectiveness of incentives and competition in schools. *Journal of Public Economics*, pag 189-213, 2006
- [3] Garcia, G., Fernandez, J. 2009. A course on Algorithms and Data Structures Using on-line Judging. ITiCSE'2009. Paris, Francia.
- [4] Garcia, G. and Fernandez, J. Make Learning Fun with Programming Contest. Department of Informatics and Systems. University of Murcia
- [5] Guerreiro, P. and Georgouli, K. 2008. Enhancing Elementary Programming Courses Using E-learning with a Competitive Attitude. *International Journal of Internet Education*. Vol. 10.
- [6] Heintz, F. and Färnqvist, F. 2012. Pedagogical Experiences of Competitive Elements in a Algorithms Course. *L THS 7:e Pedagogiska Inspirationskonferens*.
- [7] Hernandez, B., Montaner, T., Sese, J. and Urquizu, P. The role of social motivations in e-learning: How do they affect usage and success of ICT interactive tools? *Computers in Human Behavior*.
- [8] Leal, J. and Silva, F.: Using Mooshak as a Competitive Learning Tool, Universidad de Porto. Portugal
- [9] Massey, B. Experience With A Process For Competitive Programming. Computer Science Department. Portland State University.
- [10] Revilla, M. and Manzoor, S. 2008. Competitive Learning in Informatics: The Uva Online Judge Experience. *Olympiads in Informatics*, Vol 2, 131-148.
- [11] Yengin, I., Karahoca, D., Karahoca, A. and Yucel, A. Roles of teachers in e-learning: How to engage students & how to get free e-learning and the future. *Procedia Social and Behavioral Sciences*.