

Medición del trabajo del estudiante como medio para mejorar el proceso de enseñanza-aprendizaje de la programación

Fredi Palominos Villavicencio
Universidad de Santiago de Chile
Avda. Las Sophoras 175, Estación
Central, Santiago, Chile.
+56227182045
fredi.palominos@usach.cl

Israel Leiva Aburto
Universidad de Santiago de Chile
Avda. Las Sophoras 175, Estación
Central, Santiago, Chile.
israel.leiva@usach.cl

Cristóbal Leiva Aburto
Universidad de Santiago de Chile
Avda. Las Sophoras 175, Estación
Central, Santiago, Chile.
cristobal.leiva@usach.cl

RESUMEN

El aprendizaje de habilidades algorítmicas y de programación en los estudiantes universitarios, presenta niveles de dificultad y requiere tiempos de dedicación muy heterogéneos. Para los profesores, es difícil responder a esta diversidad, por cuanto las instancias de interacción con los estudiantes, generalmente están limitadas a las salas de clases o los recintos de laboratorio. Por otra parte, las estadísticas de rendimiento observables a través de las evaluaciones tradicionales, están lejos de ser los ideales para guiar óptimamente el proceso de enseñanza aprendizaje, así como para entregar un reforzamiento más personalizado.

En este trabajo se presenta un innovación, que tiene por objetivo implementar a través de un Contest Management System, actividades de ejercitación para ser desarrolladas en o fuera del aula, de modo que por medio de indicadores proporcionados por la plataforma, los cuales reflejan la eficacia y si corresponde, la eficiencia de la respuesta de cada estudiante, permita monitorear y retroalimentar oportunamente el estudio y la ejercitación de los alumnos.

Mediante el seguimiento continuo de los indicadores de cada estudiante o del grupo completo, el profesor puede ajustar la actividad de curso y dirigir más adecuadamente las actividades de estudio personal de cada estudiante.

ABSTRACT

Algorithmic learning skills and programming for college students, has levels of difficulty and requires a very heterogeneous time dedication. For teachers, it is very difficult to respond to this diversity, because instances of interaction with students, usually limited to the classroom or laboratory enclosures and typical indicators of learning, observable through rates traditional evaluations are far from ideal to deliver a more personalized reinforcement.

This paper presents an innovation that aims to implement through a Contest Management System, exercise activities to develop in the laboratory or independently outside the classroom, so as to allow timely feedback monitoring and exercising the students presented by means of indicators provided by the platform, which reflect the effectiveness and, if applicable, the efficiency of the response of each student.

Through continuous monitoring of the indicators for each student or the entire group, the teacher can adjust the course activity and properly direct the study activities of students.

Categorías y descriptores de materia

K.3 Computers and Education. K.3.1 Computer Uses in Education: Computer-managed instruction (CMI), Distance learning.

Palabras clave

Enseñanza, Programación, Tecnología, Educación Superior.

1. INTRODUCCIÓN

La programación no es una tarea simple, requiere que quien programa, utilice conjuntamente una jerarquía de múltiples competencias [4] necesarias para la adecuada articulación de los tipos de constructos relativos a un paradigma. Los problemas en el aprendizaje de la programación, van mucho más allá de la mera dificultad de la materia y requiere de un fuerte trabajo personal del estudiante [6]. A lo anterior, frecuentemente debe sumarse la carencia de aptitudes de los mismos [4] y factores motivacionales, que complejizan enormemente la tarea de la enseñanza y afectan negativamente las percepciones de los estudiantes respecto de las asignaturas de programación.

El desarrollo de habilidades algorítmicas es vital para la adquisición de buenas prácticas de programación [9]. A principios de la primera década de este siglo, esta era una preocupación importante [22] y se consideraba una materia que se seguía gestando, dado que sus métodos estaban todavía en vías de consolidación [12]. Hay quienes actualmente consideran, por ejemplo, que la formación en ingeniería, requiere profundos conocimientos informáticos, de modo que los profesionales sean capaces de construir software de alta calidad “con el fin de resolver de manera óptima los problemas de su competencia” [24]. Esta preocupación ha mantenido su vigencia en el tiempo [25].

Las investigaciones coinciden en la importancia de la ejercitación personal, lo que se explica en el hecho que el aprendizaje de la programación requiere la articulación interna de una serie de

habilidades, que son de niveles muy superiores al simple conocimiento enciclopédico [11].

En las carreras del ámbito de la computación, los cursos de programación avanzados, requieren que los estudiantes adquieran competencias asociadas a la aplicación de diferentes paradigmas de programación y en particular, se busca y se valora la eficiencia de las soluciones.

El aprendizaje de habilidades algorítmicas y de programación en los estudiantes universitarios, presenta niveles de dificultad muy diversos y requiere tiempos de dedicación muy heterogéneos. Para los profesores es difícil responder a esta diversidad, por cuanto las instancias de interacción con los estudiantes, generalmente están limitadas a las salas de clases o los recintos de laboratorio y el profesor conoce muy poco sobre el trabajo del alumno fuera del aula. En este contexto, el aprendizaje y en particular, el nivel de logro de las competencias algorítmicas, permanece en un nivel crítico [18][19] y las evaluaciones sólo aportan información parcial, en algunos momentos específicos del desarrollo de cada curso, la cual no está necesariamente desagregada respecto de los conceptos y temas claves en las asignaturas. Por lo tanto, es necesario que los profesores sean extremadamente cuidadosos en el diseño del material educativo [6] [12] [14] [20] y deben preocuparse de mostrar casos de estudio interesantes [5].

Finalmente, las estadísticas de rendimiento observables a través de los tipos de evaluaciones tradicionales, están lejos de ser los ideales para guiar óptimamente el proceso de enseñanza aprendizaje así como para entregar un reforzamiento más personalizado.

2. LA ENSEÑANZA DE LA PROGRAMACIÓN EN LA EDUCACIÓN SUPERIOR

2.1 Contexto en que se realiza la enseñanza

Si bien los estudiantes, al ingreso a la educación superior suelen estar bastante familiarizados con el uso de computadores, carecen de las habilidades algorítmicas y conocimientos de programación, que posteriormente serán exigidos por el currículum de muchas carreras universitarias.

Durante el proceso de enseñanza aprendizaje, surgen en los estudiantes diferentes síntomas o problemas, entre los que se cuentan: (a) Niveles de motivación muy heterogéneos, (b) Diferencias en los ritmos de aprendizaje, (c) Diferencias en las aptitudes de los estudiantes, (d) Utilización de métodos de estudio no apropiados para la asignatura, (e) Interacción profesor-alumno menor a la realmente necesaria, (f) Limitaciones para retroalimentar el trabajo personal de los estudiantes, etc.

En este contexto, es muy difícil para el profesor tener claridad, en cada momento del curso, respecto de la profundidad y calidad del aprendizaje logrado por cada estudiante. Los instrumentos de evaluación, suelen medir las habilidades algorítmicas de forma combinada, por lo que las calificaciones no reflejan claramente el nivel de logro de cada una de ellas.

A través de guías y otros medios, los profesores proponen ejercicios prácticos, sin embargo, es difícil conocer el trabajo real desarrollado por cada estudiante. En consecuencia, la falta de información respecto a la evolución del aprendizaje de cada

estudiante, les impide hacer un acompañamiento pedagógico más personalizado.

Producto del estilo, características y condiciones en que se realiza la docencia universitaria, la interacción entre profesores y estudiantes se produce en una cantidad y oportunidad que está por debajo de lo recomendable, generando los siguientes problemas: (a) limitaciones para orientar eficazmente el aprendizaje de los estudiantes, (b) retroalimentación escasa y tardía, (c) inseguridad de los estudiantes, (d) comunicación preferentemente unidireccional (desde el profesor hacia el estudiante), (e) serias limitaciones para hacer una enseñanza más personalizada, etc.

En consecuencia, además de las dificultades propias de la programación, muchos estudiantes experimentan una percepción de inseguridad, que afecta el aprendizaje y en diversas ocasiones, gatilla el fracaso de alumno en la asignatura.

En consecuencia, la enseñanza de la programación en la educación superior, se realiza en un contexto que no provee las condiciones más favorables para obtener resultados de aprendizaje óptimos. A esto se suma que, en su mayoría, los jóvenes estudiantes no tienen aprendizajes previos de programación [24]. Si bien, actualmente los problemas relacionados con el aprendizaje de la programación son conocidos, están lejos de estar resueltos [1] [11] [4] [6].

2.2 Las plataformas de apoyo a la docencia y la retroalimentación

La utilización de plataformas de apoyo a la docencia, si bien constituyen un aporte relevantes, no aminoran la necesidad de que el profesor realice una planificación cuidadosa y seleccione un conjunto de recursos multimediales adecuadamente diseñados [2] [8] [20]. En este contexto, la retroalimentación se destaca como un elemento fundamental en el aprendizaje de los estudiantes [17]. En principio, la retroalimentación suele ser abundante en aspectos de baja dificultad (como la sintaxis), pero escasea en los momentos que los estudiantes se enfrentan a problemas complejos, que requieren estrategias sofisticadas [32].

La utilización de las TIC ha permitido generar nuevas instancias y oportunidades de retroalimentación, superando las limitaciones temporales y de comunicación [16]. No obstante, la mayoría de las herramientas no resuelven el problema derivado de la falta de disponibilidad de tiempo del profesor, siempre inferior a las necesidades de retroalimentación reales de los estudiantes. Soluciones parciales a estos problemas, sustentados en la introducción de tutores en línea o presenciales, no facilitan al profesor todo el conocimiento respecto del nivel de aprendizaje de sus estudiantes y tampoco resuelven el gran problema derivado de la retroalimentación tardía. La superación de estas falencias, requieren de la introducción de nuevos elementos, como la tecnología [16].

2.3 Los esfuerzos por innovar

Existe preocupación por la enseñanza de la programación, la que se manifiesta en diversas experiencias relativas a innovaciones metodológicas en base a tecnología [7] [1]. Existen diversas experiencias donde se ha buscado respuesta a este problema, a través de diferentes metodologías de enseñanza [2] [3] [9] [13] [14]. Se ha usado estrategias de trabajo colaborativo [23], resolución de problemas [26], modelos de orientación visual [27], utilización inicial de pseudocódigo como un medio para hacer un

transición gradual a los lenguajes de programación [28], ambientes de aprendizaje especialmente diseñados para la enseñanza de la programación [8] [9] y otros [29] [30]. Diversas experiencias han tenido resultados positivos en contextos específicos, sin embargo, no existe consenso en términos de cuál es el mejor método para enseñanza programación [31]. Tampoco debe olvidarse, que todas las personas no piensan ni aprenden de la misma manera [18] [19], por lo que es muy difícil llegar a una metodología que se adapte a todos los tipos de estudiantes.

3. MÉTODO

La innovación presentada en este trabajo, corresponde a una investigación de tipo descriptiva, con un diseño cuasi experimental, que aborda la necesidad de mejorar la enseñanza de la programación, a través de la introducción de herramientas que permiten obtener más información sobre los logros en el aprendizaje de los estudiantes, durante la realización de actividades prácticas en aula o actividades personalizadas, focalizadas en las debilidades de cada estudiante.

La metodología consiste en el desarrollo de actividades prácticas (formativas y sumativas), a los cuales se accede vía un Contest Management System (CMS), que incluyen distintos tipos de ejercicios,. Algunas de las actividades se realizan en el aula y corresponden principalmente a trabajo individual de cada estudiante. Para cada uno de los ejercicios de las actividades, se registra el número de intentos que cada estudiante realiza hasta obtener una respuesta correcta, así como el tiempo que invierte en desarrollarla, el número de ejercicios que realiza y en general, la efectividad que muestra en las soluciones desarrolladas. En función de dichos resultados, el profesor puede conocer respecto de cada estudiante, indicadores respecto de su nivel de habilidad en la resolución de problemas, para recomendar un trabajo más personalizado.

Las actividades serán continuamente monitoreadas por el profesor de la asignatura y a partir de los resultados de las mismas, se introducirán cambios en los procesos pedagógicos de la asignatura (en y fuera del aula) y se regulará el trabajo personal de los estudiantes, en la cantidad y frecuencia que cada uno lo requiera.

La hipótesis central de esta innovación radica en la utilización de indicadores de desempeño que se derivan de las actividades que desarrollan los estudiantes y profesores en el CMS. Esta información permitirá mejorar significativamente la información disponible para el profesor y estudiante, en los siguientes aspectos:

- Proporcionará al profesor información directa respecto a cómo cambia la capacidad de cada estudiante, para resolver distintos tipos problemas, en diversos momentos del curso.
- Permitirá al profesor asignar a sus estudiantes actividades personalizadas acordes a sus resultados de aprendizaje.
- Proporcionará al estudiante retroalimentación inmediata respecto de las soluciones desarrolladas.
- Permitirá al estudiante conocer su ranking respecto de otros compañeros de su clase.

A través de estos indicadores, el profesor y cada estudiante podrán conocer el nivel de competencia alcanzado respecto de cada una de las habilidades algorítmicas requeridas por la asignatura, en diferentes momentos del desarrollo del curso. Además, los

profesores podrán objetivamente determinar qué tipo de ejercicios contribuyen de forma más efectiva al aprendizaje de sus estudiantes.

3.1 Plataforma de apoyo

Con motivo de la Olimpiada Internacional de Informática realizada el año 2012, se presentó el software Contest Management System (CMS) [15], con el objetivo de proporcionar un sistema de clasificación seguro, robusto, desarrollado para la comunidad, extensible, fácilmente adaptable y utilizable, que se adaptase de forma natural a las necesidades de la competencia. Si bien esta herramienta no fue concebida con fines pedagógicos, actualmente se está adaptando para aprovechar su funcionalidad como una potente herramienta de apoyo al proceso de enseñanza aprendizaje [21].

Una consecuencia inmediata del uso de dicho CMS, es una forma de retroalimentación inmediata y automática y el alumno puede verificar de forma autónoma la eficacia y eventualmente la eficiencia, de la solución desarrollada. Esta solución además permite disminuir en parte los costos derivados de la necesidad de recursos humanos asociados al trabajo de tutores, en línea o presenciales.

CMS está organizado de manera modular, con diferentes servicios que se ejecutan paralelamente en diferentes máquinas. Dichos servicios proveen interfaces que permiten la comunicación entre ellos y en algunas casos, aplicaciones web para administradores y competidores. Además, los servicios pueden ser iniciados y detenidos de manera independiente, sin afectar al resto del sistema. Si la base de datos que soporta el sistema funciona adecuadamente, es posible reemplazar rápidamente alguna máquina o software fallido, con una copia idéntica, sin preocuparse por pérdida de datos o de transferir información entre la máquina antigua y la nueva copia.

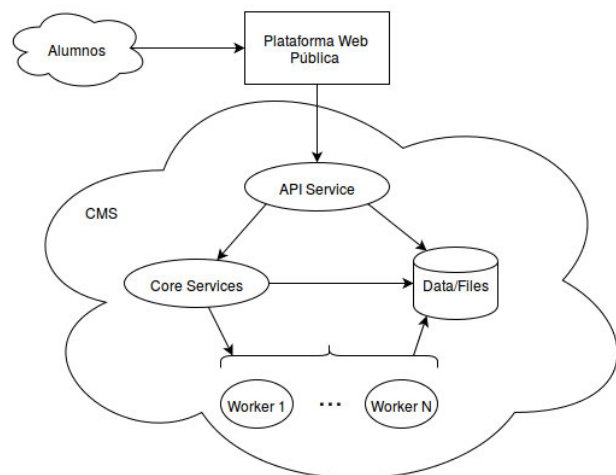


Figura 1: Organización general del sistema.

En el contexto de este trabajo [21] se ha desarrollado la Plataforma de Ejercicios de Programación (PEP), cuyo objetivo es extender CMS para poder utilizar sus recursos en el proceso de enseñanza-aprendizaje de asignaturas de programación.

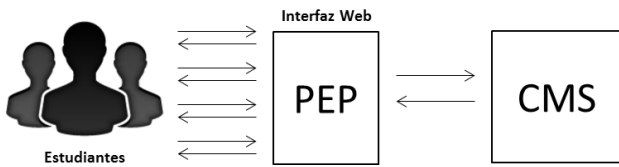


Figura 2: Interacción entre PEP, el CMS y los usuarios.

Como se presenta en la figura 2, la interfaz web denominada PEP, que está siendo desarrollada en PHP y Python, es quien proporciona todos los servicios de acceso a los estudiantes y profesores, e intermedia el trabajo con el CMS. Todas las funcionalidades relacionadas con el monitoreo del proceso de enseñanza-aprendizaje están implementadas en PEP. De la misma forma, los ejercicios, las actividades que se construyen en torno a los ejercicios, así como los indicadores de uso y efectividad, son registrados por PEP, en una base de datos implementada en MySQL.

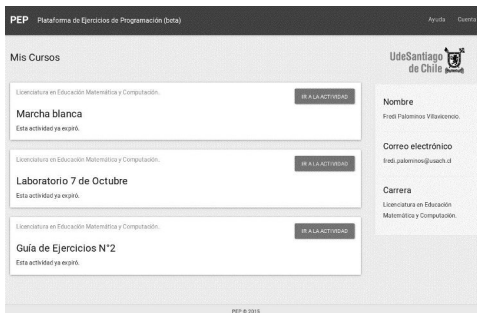


Figura 3: Interfaz de PEP, acceso a los cursos.

De este modo, se pone a disposición de los profesores, una plataforma que permite presentar a los estudiantes actividades constituidas por diversos conjuntos de ejercicios de programación, para ser desarrollados dentro de una determinada ventana de tiempo. Para cada ejercicio, el alumno puede descargar el enunciado, subir un código y obtener feedback automático respecto a la eficacia/eficiencia de su solución. En el intertanto, la plataforma web se comunica internamente con CMS a través de un servicio API desarrollado específicamente para este proyecto, el cual se encarga de traducir toda la comunicación desde la plataforma web hacia los principales componentes de CMS (ver Fig. 1). Esto ha permitido generar una capa de abstracción que reemplaza las competiciones por actividades de asignaturas, sin necesidad de realizar grandes cambios en el funcionamiento interno de CMS (Core Services).

4. PRIMEROS RESULTADOS

Actualmente, la metodología está siendo sometida a una prueba experimental, con un grupo de 43 estudiantes de primer año de la carrera de Pedagogía en Matemática y Computación, del Departamento de Matemática y Ciencia de la computación de la USACH. Dado que los 43 estudiantes corresponden al total de los nuevos estudiantes del primer año de la cohorte 2015, no ha sido necesario realizar un plan de muestreo.

En esta primera prueba, se han realizado actividades de inducción, las que se desarrollaron en una sala donde cada alumno cuenta con un computador (laboratorio).

Algunas actividades son formativas y otras sumativas. Cada actividad contempla entre dos y cuatro problemas, los cuales los alumnos resuelven individualmente en su computador, y la someten a prueba en el CMS, cuando consideran que tienen una solución correcta. La plataforma prueba cada solución en base a diversos casos de prueba, destinado a detectar situaciones que los alumnos deberían haber contemplado en su solución. El sistema responde informando si la solución es correcta y mediante un sistema de puntajes, el porcentaje de casos de prueba que la solución ha resuelto correctamente.

En el caso de las actividades sumativas, los alumnos deben responder en un tiempo determinado, durante la duración de la clase. Aquellos estudiantes que no alcanzan a realizar la actividad en el tiempo permitido, podrán completar la actividad posteriormente por su cuenta, pero esta situación pesará en la calificación final y se exigirá que la respuesta final sea coherente con el trabajo desarrollado en clases.

A modo de ejemplo, los resultados preliminares obtenidos en una actividad de carácter sumativo, que consistió en un ejercicio relativo a recursividad y un segundo ejercicio relativo a uso de vectores unidimensionales, son los siguientes:

Problema de recursividad:

- 1) Sólo el 35,5% de los estudiantes logró someter a prueba una solución al problema en el tiempo permitido. El 64,5% restante, debió entregar la solución posterior al término de la clase.
- 2) El 25,8% de los alumnos realizó un intento, el 9,7% hizo dos intentos y el 3,2% realizó tres.
- 3) Sólo el 12,9% de los estudiantes del curso realizó una solución que resolvió correctamente todos los casos de prueba.

Problema de arreglos unidimensionales:

- 1) Sólo el 45,2% de los estudiantes logró someter a prueba una solución al problema en el tiempo permitido. El 54,8% restante, debió entregar la solución posteriormente al término de la clase.
- 2) El 32,3% de los alumnos realizó un intento, el 3,2% hizo dos intentos y el 9,7% realizó tres o más intentos.
- 3) A diferencia del problema relativo a recursividad, el 38,7% de los estudiantes del curso realizó una solución que resolvió correctamente todos los casos de prueba.

5. CONCLUSIONES

Los resultados obtenidos hasta la fecha, indican que la implementación de una plataforma como la presentada en este proyecto, tiene una relación costo/beneficio que justifica su utilización como apoyo al proceso de enseñanza-aprendizaje. La información generada por el sistema permite tener mucho mayor conocimiento respecto del nivel de logro de las habilidades algorítmicas por parte de los estudiantes, así como el tiempo que les requiere concretar soluciones efectivas.

Si bien, en la etapa inicial, la implementación de actividades demanda mayor tiempo de dedicación para el profesor, con el correr del tiempo permitirá construir conjunto de ejercicios, que podrán ser reutilizados en diferentes periodos. De esta forma, el esfuerzo inicial de desarrollo será progresivamente sustituido por la ventaja de contar con ejercicios probados, que aportan mucha mayor información para la planificación pedagógica y el ejercicio de la docencia en la asignaturas de computación.

Si bien los resultados y la información generada es parcial y preliminar, el sistema ya está aportando información relevante, que de otra manera es muy difícil medirla sin la ayuda e intervención de terceras personas. En efecto, la inversión de tiempo de trabajo y los recursos humanos necesarios para hacer un seguimiento de este tipo con medios tradicionales, hacen viable este tipo de estudios sólo en condiciones experimentales y no en el ejercicio cotidiano de la enseñanza de la programación.

Finalmente, una vez construida la primera versión completa de la plataforma, será necesario abordar los aspectos didácticos relacionados con el uso de la herramienta, por parte de los profesores y estudiantes. Además, se debe contemplar el perfeccionamiento de los profesores en el uso de la misma, delimitando sus potencialidades y limitaciones.

6. AGRADECIMIENTOS

Proyecto de Innovación Docente Evaluación del efecto de actividades prácticas focalizadas realizadas a través de un CMS, en los niveles de logro de habilidades algorítmicas en cursos de programación de computadores (2015-2016), Dirección de Pregrado, Vicerrectoría Académica, Universidad de Santiago de Chile.

7. REFERENCIAS

- [1] Kirsti **Ala-Mutka**, Problems In Learning And Teaching Programming, A literature study for developing visualizations in the Codewitz-Minerva Project, Institute of Software Systems, Tampere University of Technology, Finland, 2009.
- [2] Ozgur **Aktunc**, A Teaching Methodology for Introductory Programming Courses using Alice, Department of Engineering, St. Mary's University, San Antonio, United States, International Journal of Modern Engineering Research (IJMER), Vol.3, Issue.1, Jan.-Feb. 2013 pp-350-353.
- [3] Micaela **Esteves**, Benjamim Fonseca, Leonel Morgado and Paulo Martins, Improving teaching and learning of computer programming through the use of the Second Life virtual world, British Journal of Educational Technology, 2010.
- [4] Tony **Jenkins**, On The Difficulty Of Learning To Program, School of Computing University of Leeds Leeds, UK., 3rd Annual LTSN-ICS Conference, Loughborough University © 2002 LTSN Centre for Information and Computer Sciences
- [5] Avinash **Kak**, Teaching Programming, Purdue University, July 2008.
- [6] Essi **Lahtinen**, Kirsti Ala-Mutka, Hannu-Matti Järvinen, A Study of the Difficulties of Novice Programmers, ITiCSE'05, Monte de Caparica, Portugal, June 27–29, 2005.
- [7] **Llorens Largo**, Faraón, Satorre Cuerda, Rosana, Enseñar Programación en las Ingenierías Informáticas, II Jornadas Nacionales de Innovación en las Enseñanzas de las Ingenierías, Instituto de Ciencias de la Educación, Universidad Politécnica de Madrid. Madrid, 3 y 4 de diciembre de 1996.
- [8] P. **Montero** Lagos y otros, Ambientes y Recursos de Aprendizajes Multimediales: Avances y Proyecciones de una Experiencia en la Universidad de Santiago de Chile, experiencias del proyectos MECESUP USA0104 (compilación), Ed. CEDETEC, USACH, 2008.
- [9] Fredi **Palominos** V., Rosa Barrera C., Patricio Montero L.. Diseño y construcción de un ambiente de aprendizaje para la enseñanza de algoritmos y estructura de datos en la enseñanza superior universitaria, Congreso Chileno de Educación Superior en Computación, Jornadas Chilenas de Computación, UNAP, Noviembre de 2007.
- [10] Marc R. **Prensky**, Teaching Digital Natives: Partnering for Real Learning, ISBN-13: 978-1412975414, Corwin; Edición: New, 2010.
- [11] **Renumol** V. G., Jayaprakash S., Janakiram D., Classification of Cognitive Difficulties of Students to Learn Computer Programming, Department of Computer Science, Indian Institute of Technology, Madras, India, Technical Report, 2009.
- [12] Ariel Ferreira Szpiniak, Guillermo A. **Rojo**, Enseñanza de la programación, TE&ET | Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, , Vol. 1, N° 1, Dic. 2006.
- [13] Josep **Soler**, Ferran Prados, Jordi Poch e Imma Boada, ACME: Plataforma de Aprendizaje Electrónico (e-learning) con Funcionalidades Deseables en el Ámbito de la Ingeniería, Formación Universitaria, Vol. 5(3), 3-16, 2012.
- [14] Raúl A. **Trejo**, Rubén D. Santiago, Lourdes Quezada y Francisco Delgado. La Enseñanza De La Programación Dentro De Un Modelo De Integración Curricular: La Experiencia Del Proyecto Principia. Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Edición: Educación Superior y Nuevas Tecnologías, Rede Iberoamericana de Informática Educativa, 2002.
- [15] Stefano Maggiolo, Giovanni Mascellani, Introducing CMS: A Contest Management System, Olympiads in Informatics, Vilnius University, Vol. 6, 86–99, 2012.
- [16] Redecker, C., y Johannessen, Ø Changing Assessment — Towards a New Assessment Paradigm Using ICT.. European Journal of Education, vol. 48, n°1, 79-96, 2013..
- [17] Larraguibel Q., E. y Chávez R., J. El feedback como herramienta de evaluación formativa y formadora en educación universitaria. Reporte técnico actualmente sometido a evaluación en revista especializada, 2015.
- [18] Díaz, Hernán; Cañete, Lucio; Costa, Cristián; Palominos, Fredi; Córdova, Felisa, Neurotechnologies for Education Improvement: Self-Knowledge After opening the Black Box, Journal Plus Education, ISSN: 1842-077X, E-ISSN (online) 2068 – 1151, Vol VIII (2012), No. 2, pp 44 – 52, 2013.

- [19] Nazre Abdul Rashid, Mohd. NasirTaib, SahrimLias, NorizamSulaiman, ZunairahHj. Murat, RosShilawani S. Abdul Kadir, Learners' Learning Style Classification related to IQ and Stressbased on EEG, International Conference on Education and Educational Psychology (ICEEPSY), Procedia - Social and Behavioral Sciences 29, pp. 1061 – 1070, 2011.
- [20] C. Coll, Psicología de la educación y prácticas educativas mediadas por las tecnologías de la información y la comunicación, Sinéctica, vol. 25, pp. 1–24, 2004.
- [21] Palominos, F., Barrera,R., Fortalecimiento de las habilidades algorítmicas y de programación de los estudiantes de las carreras de pregrado del Departamento de Matemática y Ciencia de la Computación, Proyecto de Innovación Docente, Vicerrectoría Académica, USACH, 2015.
- [22] Costelloe, E. (2002). Teaching Programming - The State of the Art. Recuperado 26 de junio de 2015.
- [23] Teague, D., & Roe, P. (2008). Collaborative learning: towards a solution for novice programmers. En Proceedings of the tenth conference on Australasian computing education-Volume 78 (pp. 147–153). Australian Computer Society, Inc.
- [24] Aspectos Fundamentales para al enseñanza de programación básica en ingeniería. (2010). Revista Avnaces en Sistemas e Informática, 7(1), 8–14.
- [25] Cuerda, R. S., Largo, F. L., & García, J. A. P. (s. f.). Enseñar Programación en las Ingenierías Informáticas.
- [26] De Raadt, M. (2007). A review of Australasian investigations into problem solving and the novice programmer. Computer Science Education, 17(3), 201–213.
- [27] Shehane, R., & Sherman, S. (2014). Visual teaching model for introducing programming languages. Journal of Instructional Pedagogy Volume.
- [28] PSeInt. Recuperado 19 de agosto de 2015, a partir de <http://pseint.sourceforge.net/>
- [29] Cuevas Valencia, R. E., Bautista Jiménez, H., & Medina Martínez, J. C. (2013). Propuesta metodológica para la enseñanza de la programación en la Unidad Académica de Ingeniería de la UAGro. Revista Vínculos, 10(2).
- [30] Vihavainen, A., Paksula, M., & Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. En Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 93–98). ACM.
- [31] Michaelson, G. (2015). Teaching-Programming-with-Computational-and-Informational-Thinking. Journal of Pedagogic Development, 5(1).
- [32] Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. En Proceedings ascilite Singapore