

Innovación en la Enseñanza Inicial de la Programación

Inés Friss de Kereki
Universidad ORT Uruguay
kereki_i@ort.edu.uy

Alejandro Adorjan
Universidad ORT Uruguay
adorjan@ort.edu.uy

Resumen: La programación es una de las áreas centrales en las ciencias de la computación y muchas veces se constata que en los cursos iniciales de esta materia hay alta deserción y reprobación. En el curso de Programación I de la Facultad de Ingeniería de la Universidad ORT Uruguay incorporamos un conjunto de actividades para fomentar el aprendizaje activo y la clase invertida ("flipped class") con la finalidad de mejorar los resultados de los cursos y disminuir la deserción. Los datos preliminares de la investigación muestran mejoras en los resultados del curso así como en la disminución en los índices de deserción.

Palabras clave: Programación, Aprendizaje Activo, Clase Invertida.

Introducción

La estructura de este trabajo es la siguiente: primero se presentan aspectos relacionados a la enseñanza universitaria de la programación y algunos problemas que se detectan en cuanto a sus resultados. También se incluyen reflexiones en cuanto a la enseñanza universitaria en general. Se detallan las principales características del curso de Programación I de la Facultad de Ingeniería de la Universidad ORT Uruguay. Luego, como continuación de trabajos referidos a dicho curso se presenta un conjunto ampliado de actividades innovadoras. Finalmente se presentan resultados iniciales y se ofrecen conclusiones.

Acerca de la enseñanza

La programación es el foco central de las ciencias de la computación siendo los resultados algunas veces desalentadores [16]. Es una habilidad muy útil, pero aprender a programar es difícil [24]. Es común esta concepción de dificultad [6]. Los cursos introductorios de programación presentan varios desafíos, consecuencia de la alta tasa de deserción y abandono [4].

La disminución de tasas de retención y los cambios en los estilos de aprendizaje de los estudiantes son los principales temas que actualmente enfrenta la comunidad académica del área [21]. Varias estrategias han sido adoptadas por diferentes institutos, en la organización y dictado de los cursos [4]. Por ejemplo, incluir el uso de juegos para motivar a los alumnos y disminuir la deserción [5]. Si bien variados enfoques han sido presentados, no hay un consenso aún en cuál es la manera más efectiva de enseñar a programar [28].

Una visión errónea de los índices de repetición puede tener implicaciones serias en la calidad de los cursos introductorios de programación, en el cual un docente con un alto índice de repetición en el dictado puede asumir que esto sea habitual y, consecuentemente, no tomar ninguna acción para intentar reducirlos [6]. El atraer y retener a los estudiantes implica la participación y el disfrute a nivel individual,

de equipo y de clase [27]. Las prácticas de enseñanza en el área de la programación inicial han sufrido un cambio a nivel mundial, pasando de la enseñanza centrada en el docente hacia la enseñanza centrada en el estudiante [19].

Generar espacios de reflexión sobre la enseñanza en áreas de tecnología e ingeniería específicamente fue el eje del “Programa para el fortalecimiento de la enseñanza y aprendizaje en el área de Ciencia, Tecnología, Matemática e Ingeniería” (“Program of Strengthening Teaching and Learning in the STEM Fields”), de una semana de duración, que tuvo lugar en julio de 2012 en Cambridge, Massachusetts, USA, en el marco de un convenio entre las facultades de Ingeniería de Uruguay y LASPAU-Harvard University [22]. El programa proporcionó a los docentes variadas estrategias, técnicas y elementos teóricos para aplicar e innovar en las propias clases.

En especial, los conceptos allí presentados por Ambrose (expuestos en [3]) acerca de reflexionar sobre que el conocimiento previo de los estudiantes puede entorpecer los nuevos aprendizajes, siendo esta una idea relevante al momento de diseñar e implementar actividades. También el hecho de establecer reglas claras para el funcionamiento en discusiones y otros momentos de la clase da pautas para la aplicación de actividades.

Así mismo, se remarcó la idea de fomentar el aprendizaje activo, la instrucción por pares y el uso de “concept tests”, que refieren a preguntas conceptuales diseñadas para exponer dificultades comunes en la comprensión de materiales, según presentó Mazur [17]. La instrucción por pares promueve la interacción en clase y se centra en los “concept tests” [29].

También se discutió en ese espacio de reflexión el concepto de “clase invertida” (“flipped classroom”), modelo en cual se transfieren determinados procesos fuera de la clase y dentro de ella se utiliza el tiempo para discutir conceptos y ejercicios con los estudiantes. Ampliando en este punto, la pedagogía de “flipped classroom” ha impactado significativamente en los círculos académicos universitarios en los últimos años [10]. Esta reinención de los cursos tradicionales permite a los estudiantes combinar un entorno de aprendizaje a distancia conjuntamente con la asistencia a un curso tradicional [10]. La “flipped classroom” invierte el método tradicional de enseñanza-aprendizaje, permitiendo a los estudiantes aprender a su propio tiempo y ritmo. A su vez, permite al docente mejorar la interacción y colaboración al realizar en el aula un seguimiento de cada aprendizaje individual [18].

En resumen, teniendo en cuenta las dificultades detectadas en la enseñanza y el aprendizaje de la programación así como el interés de utilizar otras formas de enseñar en general, entendemos que es útil continuar en la búsqueda y aplicación de nuevos elementos.

Programación I en Universidad ORT Uruguay

El objetivo de la materia Programación 1 de la Facultad de Ingeniería de la Universidad ORT Uruguay es iniciar la enseñanza de la programación empleando fundamentalmente técnicas de programación orientada a objetos. El énfasis está en la enseñanza de una metodología de resolución de problemas y capacitar al estudiante para desarrollar aplicaciones sencillas con lenguajes orientados a objetos. Al final del curso el estudiante estará preparado para analizar situaciones simples, diseñar una posible solución e implementarla basándose en un enfoque orientado a objetos [13]. En la Tabla 1 se presenta el contenido resumido del curso, por semanas:

Tabla 1 - Desarrollo temático del curso

Semana	Desarrollo temático del curso
1-3	Variables, estructuras de control, seudocódigo.
4	Presentación de clases y objetos, usos de clases estándar.
5-8	Creación de clases, alias, relaciones.
9	Herencia, Clases y Objetos.
10-12	Colecciones, excepciones, ordenación y búsqueda.
13	Enumeración.
14-15	Manejo avanzado de colecciones.

Los contenidos detallados son:

- Introducción a conceptos básicos. Computadores, Hardware y Software. Ciclo de desarrollo: análisis, diseño, implementación, prueba. Algoritmo, Programa. Compiladores e Intérpretes. Lenguajes. Descomposición lógica de problemas. Descomposición funcional. Descomposición modular. Estructuras de control.
- Conceptos Básicos de Orientación a Objetos: Clases, objetos. Mensajes, métodos. Relaciones entre clases y entre objetos. Herencia, polimorfismo. Asociación, Agregación. Notación. Resolución de problemas simples con Orientación a Objetos.
- Introducción a un lenguaje de programación orientado a objetos: Ambiente de programación, sintaxis, objetos simples, tipos de mensajes, tipos de variables. Definición de clases y métodos; relaciones entre clases y entre objetos, herencia y polimorfismo. Jerarquía de Clases. Colecciones. Excepciones. Resolución de problemas simples.

En particular, se utiliza Java como lenguaje de Programación durante 16 semanas, incluyendo cada semana 4 horas en salón de clase y 2 horas en laboratorio.

La evaluación del curso se realiza a través de dos entregas de programas (en grupo de 2 estudiantes) y un parcial individual final. La primera entrega es de 25 puntos en la semana 10 del curso y la segunda de 25 puntos en la semana 15. El parcial final (evaluación escrita) tiene 50 puntos. Para aprobar el curso se requieren 70 o más puntos y superar los mínimos de cada prueba (8 puntos en la primera entrega, 12 en la segunda y 25 en el parcial). Los estudiantes no sólo deben desarrollar las competencias necesarias para resolver las dificultades a nivel de resolución de problemas, sino que deben trabajar en equipo con el objetivo de resolver las dos entregas de programas.

Innovaciones previas en el curso de Programación I de la Universidad ORT Uruguay

Con el objetivo de mejorar los resultados de los cursos de Programación I y disminuir la deserción, hemos realizado en primera instancia varias innovaciones al curso y su correspondiente análisis que presentamos en [1] y [2]. En particular, las actividades presentadas ordenadas de acuerdo a la preferencia manifestada por los estudiantes son, en orden de mayor a menor:

1 "Concept Test": Un tema particular se divide en un conjunto de pequeñas presentaciones, cada una centrada en un punto importante y seguido de un test conceptual.

2 "Puzzle": Se plantea resolver el problema de encontrar el máximo de un conjunto de números. Se provee a los estudiantes con un puzzle, cuyas piezas son líneas de código. Además de las necesarias, se incluyen líneas adicionales. Los estudiantes deben seleccionar las líneas y construir activamente el algoritmo.

3 Masa de modelar: El objetivo es presentar los conceptos de clase, objetos, mensaje y alias a través del uso de masa de modelar y moldes de plástico (que representa la clase).

4 Infografía: Una infografía es una representación visual y gráfica donde se incluye información en forma de texto, imágenes, mapas, viñetas, diagramas, tablas, y/o, descripciones sobre un determinado tema. Se propone construir una infografía sobre "Qué es Java, cuándo surgió, cómo" y preguntas relacionadas. Los estudiantes elaboran su infografía en forma previa a la clase y luego se discuten en clase.

5 Scratch: Se utiliza Scratch [25] en las primeras semanas del curso con el fin de acercar los estudiantes a la experiencia de la programación y motivarlos.

6 Juego para modelado: Se traen varias cajas de juegos infantiles y se debe modelar la clase "Juego" a partir de características comunes detectadas en las cajas.

7 Wordle: Se solicita crear una nube de palabras utilizando una herramienta disponible en la web [30] a partir del código Java de una clase particular y analizar lo obtenido

8 Video: Se muestra un video sobre programación por pares para discutir cuál es el espíritu de ese enfoque, sus ventajas y desventajas.

9 Test del Minuto: En un minuto, los estudiantes al final de la clase responden: qué fue lo más importante que aprendió hoy y qué preguntas le surgen.

10 Rúbrica: Se presentan entregas de programas realizadas en cursos previos: uno apropiada y otra con errores. A partir del diseño de una rúbrica para evaluación elaborada en conjunto, se revisan los trabajos y se califican, justificándose la nota obtenida.

11 Herramientas: A partir de un conjunto de objetos variados, como una jabonera o una afeitadora desechable, en grupos deben seleccionar uno, dibujarlo, analizar sus características (material, funcionalidad, etc.) y formular preguntas sobre qué cosas querrían averiguar y cómo. A partir de este análisis, se establece analogía con el desarrollo del software.

Un subconjunto de esas actividades (actividades: 4, 6, 7, 10 y 11) obtuvo uno de los primeros premios en el Programa Regional de Emprendedorismo e Innovación en Ingeniería (PRECITYE) [20].

Ampliación de las innovaciones

En esta siguiente etapa, basados en las experiencias iniciales y a partir de los resultados y de las encuestas así como de las entrevistas a estudiantes y docentes, hemos decidido modificar algunas de las actividades citadas e incluir nuevas. El nuevo conjunto lo aplicamos desde agosto de 2014. Los cambios y agregados son:

- a. Ampliación de la infografía
- b. Inclusión de "La tapa del libro"
- c. Herramientas para "concept test"
- d. Uso de canción y música para presentar el manejo de parámetros y reuso de código
- e. Juego de roles

Además se propuso la eliminación de la actividad "Herramientas".

O sea, el curso actual contiene las primeras 10 actividades presentadas y los ajustes que proponemos. Detallaremos cada una de las incorporaciones y/o cambios. Ellos son:

a) Ampliación de la infografía

Como se indicó, una infografía es una representación que incluye entre otros textos e imágenes. Se trata de presentar claramente las ideas o conceptos requeridos, en un formato atractivo. Los estudiantes preparan la infografía, para lo cual deben buscar y seleccionar información, detectar lo relevante y disponerlo en un formato interesante. El enfoque es invertir la clase y que tengan un rol activo.

En los distintos usos de esta herramienta hemos detectado algunas oportunidades de mejora en cuanto a la presentación y participación activa de los demás estudiantes, por ejemplo, teniendo en cuenta las recomendaciones de Ambrose para la guía de la discusión así como fomentar la vinculación de sus conocimientos previos, en relación a los contenidos que los estudiantes entienden relevantes e incluyen en su infografía [3].

Para estimular la participación de todos y que no sean sólo observadores, antes de comenzar a presentar se acuerda con los estudiantes una pauta de evaluación, o sea, fijar criterios conjuntos con los cuales se evaluarán las infografías y su presentación. Así, por ejemplo, se establece que será importante el diseño, la relevancia y completitud del contenido, la forma de exponerla u otros criterios. Cada estudiante debe completar una planilla en la que a cada presentación le otorgará de 0 a 3 puntos en cada criterio. El docente también completa su planilla. Además, se van anotando los nombres de los presentadores en el pizarrón.

Durante las presentaciones se pueden formular preguntas, ajustes y comentarios, y se fomenta la discusión. Luego que se terminan las presentaciones, se le pide a cada uno de los estudiantes que, en función de sus apreciaciones, indique cuáles fueron las dos mejores. Se van marcando en el pizarrón los "votos", incluidos los del docente, y se elige la o las mejores.

En ese momento se aprovecha a dar recomendaciones generales sobre las presentaciones vistas, por ejemplo, la forma de comunicarse, destacar fortalezas de las infografías mostradas (por ejemplo la

buenas disposición o jerarquización de temas) y oportunidades de mejora (por ejemplo el uso de letras muy pequeñas que no se pueden leer o exceso de información o de color).

Luego, el docente al ver todas las presentaciones puede haber detectado algún punto faltante o incorrecto, que pasaría a explicar en ese momento, o sea, complementa el tema si es necesario. Para cerrar la actividad, se hace un "concept test". Se reparte el test múltiple opción, cada uno lo responde en forma individual, luego lo chequea con un compañero, discutiendo las posibles diferencias para unificar respuestas. Según el tamaño del grupo se puede luego verificar de a 4. Finalmente se verifica en el grupo entero. Dado que cada estudiante ya pensó cada respuesta y la verificó una o dos veces con sus pares, en general, las respuestas finales son correctas y si hay alguna equivocada mayoritariamente, se revisa el punto en detalle, usando las ideas de la instrucción por pares [29].

b) Inclusión de "La tapa del libro"

Como una variedad de la infografía, incluimos la actividad que llamamos "La tapa del libro" [15]. ¿Qué es la tapa del libro? ¿Qué es lo que está en la tapa del libro? Es lo más importante, lo fundamental. Aquí la propuesta es que preparen previamente a la clase "la tapa de un libro" del tema que nos interesa trabajar, invirtiendo la clase. Por ejemplo, si se elige el tema "Herencia" les pedimos que creen la tapa de un libro de "Herencia". Tendrán que buscar información sobre qué es la herencia, cómo se define, y qué características tiene en Java. Otro posible tema es "Excepciones".

Se procede en forma similar, sólo que en vez de verlas en la computadora de la clase, se van pasando las tapas entre los compañeros y se van leyendo. El resto de la mecánica puede ser similar a la infografía. El docente puede elegir si incluye un "concept test" u otra actividad para completar el tema. Para el caso de herencia, se decidió traer ejemplos impresos de código y se trata de detectar posibles errores (ejemplo: intentar acceder a atributos privados desde una subclase). No sólo se trata de recordar y comprender, sino aplicar y analizar, según la taxonomía de Bloom [7].

c) Herramientas para "concept test"

Para verificar las respuestas de un "concept test" se pueden utilizar distintas herramientas. La más simple es que voten con la mano, indicando cuál es la respuesta correcta. Otra opción es usar hojas de "raspaditas" para el chequeo [12]. El estudiante (o el equipo de trabajo) debe raspar la respuesta correcta, en cuyo caso aparece un símbolo en esa posición y en caso incorrecto está un espacio en blanco. Fácilmente se puede calcular el puntaje obtenido. La propuesta realizada en clase resulta altamente motivadora para los estudiantes.

Otra alternativa es indicar la respuesta a través del uso de votadores ("clickers") [23] o herramientas web como Kahoot [14] o Socrative [26]. En nuestra experiencia, el uso de los clickers complica la dinámica de la clase, porque hay que llevar los controles, repartirlos, verificar la conexión y no es tan intuitivo su uso. Para Kahoot y Socrative se usan los propios celulares de los alumnos para votar o responder. Las herramientas web citadas permiten una funcionalidad similar (o mejor) que la de los clickers, porque muy fácilmente se pueden obtener estadísticas y hacer seguimiento de los estudiantes, son sencillas de utilizar tanto por los alumnos como por los profesores y motiva a los estudiantes utilizar su celular, según la encuesta realiza a los propios estudiantes, en la cual la amplia mayoría respondió apreciar afirmativamente su uso.

d) Uso de canción y música para presentar el manejo de parámetros y reuso de código

Algunos problemas que detectamos al corregir el primer programa obligatorio son que se repetía mucho código de ingreso y validación de datos y también que se hacía un uso escaso de parámetros. Incluimos en la clase la actividad explicada en [8] aplicada en el contexto específico del reuso de código y parámetros. Primero se muestra un video "El viejo Mc Donald" [11]. Esta canción infantil refiere a un granjero que tiene una granja con varios animales. La letra tiene una parte que se repite varias veces a través del estribillo y hay diferentes sonidos de esos animales. Luego se reparte la letra impresa y se vuelve a ver el video para observar el estribillo. Se explica cuál es su función ("evitar repetir código") y la relación con el sonido de cada animal (El estribillo recibe por parámetro el "sonido"). Esta actividad permite mostrar de una manera diferente y divertida estos conceptos abstractos de alto nivel.

e) Juego de roles

Como planteo previo, se les indica a los estudiantes que busquen y analicen fuera de clase un algoritmo determinado, por ejemplo, para ubicar la posición de un valor en un vector u ordenarlo. En la siguiente clase, se realiza esta actividad, basada en [9] y presentada en [15]. Se llevan a clase juegos de 4 fichas de colores. Se plantea la actividad concreta: desarrollar un algoritmo similar a los citados (por ejemplo, "buscar la enésima aparición de un valor en un vector"). A cada grupo de 4 estudiantes se le entrega un juego de fichas. Cada estudiante toma una ficha y debe participar en la actividad según su color (azul, verde, rojo y negro). El azul actúa de profesor y analiza el proceso el verde tiene como rol ser creativo y buscar más alternativas el rojo actúa de alumno desarrollando la actividad y el negro toma una actitud de crítica constructiva. Luego, los roles se intercambian. Se fomenta el aprendizaje activo. Los alumnos guían y realizan la actividad.

Como elemento adicional a los cambios presentados y con el objetivo de brindar retroalimentación en forma temprana al curso, se modificó la evaluación del curso a partir de agosto de 2015. Se disminuyó el puntaje de la primera entrega, pasando a valer 15 puntos. El resto se mantuvo igual. Los 10 puntos restantes se asignaron a 5 "concept tests" que deben realizarse en forma individual. Cada uno contiene 10 preguntas. Se utilizan desde las primeras semanas, a razón de uno cada 2-3 semanas. En algunas ocasiones se pidió que leyeran previamente un material y luego se aplica el "concept test" y en otras se utilizó luego de discutir el tema en clase. Cada test lleva unos 15-20 minutos de la clase. Posteriormente a su realización, se comentan las respuestas y se aclaran dudas.

Evaluación preliminar

Se compararon los últimos dos semestres (agosto-diciembre 2014, marzo-julio 2015) donde se aplicaron todas las actividades, esto es el conjunto inicial más las modificaciones y ampliaciones. En un grupo se aplicaron todas ellas y se comparó con otros dos grupos en los cuales no se usaron. En todos los casos, los docentes son experimentados. Los grupos fueron elegidos al azar y los alumnos no fueron informados sobre la experimentación. Los resultados se presentan en la siguiente tabla (Tabla 2: Resultados Finales de Aprobación):

Tabla 2 - Resultados Finales de Aprobación

Aprobación	Grupo con todas las actividades	Grupos de control sin actividades
Cantidad de estudiantes aprobados en curso de agosto-diciembre/2014	14 de 24 (58%)	a) 12 de 30 (40%) b) 10 de 29 (34%)
Cantidad de estudiantes aprobados en curso de marzo-julio/2015	15 de 27 (56%)	a) 6 de 17 (35%) b) 12 de 26 (46%)

Se observa que en los grupos que se utilizaron las actividades, los porcentajes de aprobación son más altos que en los grupos en los que no se utilizaron. Los resultados son mejores. Además, en las encuestas realizadas a los estudiantes, manifiestan que les agrada incluir actividades.

Cabe notar que, si se compara con el semestre marzo-julio de 2014, donde se aplicó en un grupo de 31 estudiantes el conjunto inicial de herramientas presentado en [1] y [2], se obtuvo como resultado 17 aprobados (54%), porcentaje algo inferior a los obtenidos al usar todas las actividades.

En relación a la deserción, en agosto de 2015 con el cambio en la evaluación en curso actualmente, podemos observar que la deserción temprana parece disminuir en los grupos en que se usan las actividades, según se aprecia en la Tabla 3: Deserción. Consideramos deserción temprana a quienes abandonan el curso antes de la primera evaluación obligatoria (entrega del primer trabajo obligatorio en la semana 10 del curso).

Tabla 3 - Deserción

Deserción temprana	Grupo con todas las actividades	Grupos de control sin actividades
Sistema de evaluación anterior (agosto-diciembre/2014)	4 de 24 (17%)	a) 3 de 30 (10%) b) 5 de 29 (17%)
Sistema de evaluación anterior (marzo-julio/2015)	5 de 27 (19%)	a) 4 de 17 (24%) b) 3 de 26 (12%)
Sistema de evaluación nuevo (con 5 concept test) (agosto-diciembre/2015)	2 de 28 (7%)	5 de 19 (26%)

Conclusiones y trabajo futuro

Es importante como docente reflexionar sobre la propia práctica e incluir variedad e innovación en las propuestas con el objetivo adicional de mantener el interés en el curso y el desafío hacia planteos novedosos, sin olvidar por supuesto el contenido curricular a tratar. Se logra así mayor involucramiento de los estudiantes, quienes toman un rol activo en su aprendizaje y se obtiene un trabajo más personalizado.

Incorporar el conjunto completo de actividades presentado en el curso de Programación I parece mejorar los resultados de los alumnos, según la experimentación realizada. También, el seguimiento temprano parece colaborar en disminuir la deserción. Se planea seguir revisando e incorporando nuevas actividades y replicar la experimentación, a los efectos de poder verificar los resultados.

Los docentes que participaron en la experimentación observaron un mayor compromiso y participación por parte de los estudiantes. Su rol como docente cambió: de ser el centro pasó a ser un moderador o articulador de la clase, y los protagonistas son los propios estudiantes. Finalmente, un punto interesante a destacar es que la mayoría de estas actividades son transversales y pueden aplicarse a otras áreas de conocimiento, lo que aumenta su alcance y potencial.

Referencias bibliográficas

- [1] Adorjan, A., "Un enfoque de Inteligencias Múltiples y Competencias aplicado a la enseñanza inicial de la Programación". Tesis de Maestría. Universidad ORT Uruguay, 2014. Accedido en octubre 2015 en: <http://bibliotecas.ort.edu.uy/bibid/79474/file/1346>
- [2] Adorjan, A., Kereki, I., "Design and Experimentation of Activities for CS1: A Competences Oriented Approach (unpacking the Informed Design Teaching and Learning Matrix)", CLEI Electronic Journal, V. 17, N. 3, 2014
- [3] Ambrose, S., Bridges, M., DiPietro, M., Lovett, M., Norman, M., Mayer, R., How Learning Works: Seven Research-Based Principles for Smart Teaching. Jossey-Bass; USA, 2010
- [4] Ambrosio, A.P.; Costa, F.M.; Almeida, L.; Franco, A.; Macedo, J., "Identifying cognitive abilities to improve CS1 outcome", Frontiers in Education Conference (FIE) 2011, Rapid City, S. Dakota, USA, 2011
- [5] Barnes T., Powell E., Chaffin A. & Lipford H, "Game2Learn: improving the motivation of CS1 students.", Proc. of the 3rd international conference on Game development in computer science education (GDCSE '08), ACM, New York, NY, USA, 2008
- [6] Bennedsen J. & Caspersen M., "Failure rates in introductory programming.", Symp. on Computer science ed. SIGCSE Bull. V39 N2, 2007
- [7] Bloom, B.S. (Ed.), Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain, Toronto: Longmans, Green, New York, 1956
- [8] CODE Studio, "Escribir canciones". Accedido en octubre 2015 en: <https://studio.code.org/unplugged/unplug7-es-ES.pdf>
- [9] de Bono, E., 6 sombreros para pensar. Ed. Paidós, España, 2012
- [10] Elliott, R., "Do Students Like the Flipped Classroom? An Investigation of Student Reaction to a Flipped Undergraduate IT Course". IEEE Frontiers in Education Conference - FIE 2014. Madrid, España, 2014
- [11] "El viejo Mc Donald" (canción). Accedida en octubre 2015 en <https://www.youtube.com/watch?v=vofWecNQaK8>
- [12] Epstein Educational Entreprises: "IF AT TEST". Accedido en octubre 2015 en <http://www.epsteineducation.com/home/testmaker/>
- [13] Facultad de Ingeniería, Universidad ORT Uruguay: Programación I. Accedido en octubre 2015 en: http://fi.ort.edu.uy/22534/5/programacion_1.html
- [14] Kahoot. Accedido en octubre 2015 en <https://create.kahoot.it/>
- [15] Kereki, I. , "Fomentando el emprendedorismo desde cursos iniciales de Programación". "IX Workshop Latinoamericano de Innovación y Emprendedorismo de la Red EmprendeSUR", www.9wemdq.com, Mar del Plata, Argentina, 2015
- [16] Kinnunen P., Malmi, L., "Why students drop out CS1 course?", Proceedings of the Second International

Workshop on Computing education research (ICER '06), UK, 2006

[17] Mazur, E., Accedido en octubre de 2015 en <http://mazur.harvard.edu/>

[18] Nai Kiat, P., Kwong, Y. T., "The Flipped Classroom Experience", CSEE&T 2014, Klagenfurt, Austria, 2014

[19] Pears A., Seidman S., Malmi L., Mannila L., Adams E., Bennedsen J., Devlin M., Paterson J., "A survey of literature on the teaching of introductory programming", ITICSE-WGR Working Group Reports on ITICSE on Innovation and Technology in Computer Science Education, USA, 2007

[20] PRECITYE: Programa Regional de Emprendedorismo e Innovación en Ingeniería: Ingenieros Emprendedores-Ejercicios. Accedido en octubre 2015 en <http://www.ingemprendedores.org/ejercicios/?id=17>

[21] Price, K., Smith, S., "Improving student performance in CS1". J. Comput. Sci. Coll. Vol 30, No. 2 (December 2014), pp 157-163, 2014

[22] "Program on Strengthening Teaching and Learning in the Science, Technology, Engineering and Mathematics (STEM) Fields - Uruguay", Cambridge, Ma, USA, 2012. Accedido en octubre 2015 en <http://laspau.org/ar2012/idia>

[23] QOMO Hite Vision: "QClick QRF500". Accedido en octubre 2015 en <http://qomo.com/Product.aspx?ProductID=88#this>

[24] Robins A., Rountree J, Rountree N., "Learning and teaching programming: A review and discussion", Computer Science Education, Vol 13, No. 2, pp 137-172, 2003

[25] Scratch. Accedido en octubre 2015 en <http://scratch.mit.edu>

[26] Socrative. Accedido en octubre 2015 en <http://www.socrative.com/>

[27] Sprint, G.; Cook, D., "Enhancing the CS1 student experience with gamification", Integrated STEM Education Conf. (ISEC), USA, 2015

[28] Vihavainen, A., Paksula, M., Luukkainen, M., "Extreme apprenticeship method in teaching programming for beginners.", Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11, USA, 2011

[29] Watkins, J., Mazur, E., "Retaining students in science, technology, engineering, and mathematics (STEM) majors", J. Coll. Sci. Teach, Vol 42, No 5, pp 36-41, 2013

[30] Wordle. Accedido en octubre 2015 en <http://www.wordle.net/>

