

Incorporación de "Kinesthetic Learning Activities" al curso de Programación I: uso y resultados

Inés Friss de Kereki

Universidad ORT Uruguay
Cuareim 1451, Montevideo, 11100, Uruguay
kereki_i@ort.edu.uy

Abstract. En este trabajo se presenta la incorporación sistemática de actividades de aprendizaje con movimientos y sensaciones (en inglés: *KLA: Kinesthetic Learning Activities*) al curso de Programación I. Se detallan las diferentes actividades y su utilización en el curso. Se evaluaron los resultados desde el punto de vista de aprobación de la materia. Se tuvo en cuenta además la opinión de los alumnos participantes. Del análisis de los resultados se observa un elevado grado de exoneración de la materia en comparación a los cursos tradicionales.

Keywords: Programación, Enseñanza, Aprendizaje, orientación a objetos, KLA

1 Introducción

Este trabajo tiene por objetivo contribuir a la mejora de los procesos de enseñanza y de aprendizaje de la Programación, en particular de la Programación Orientada a Objetos. En la Universidad ORT Uruguay, desde 1996, se enseña este paradigma en los cursos iniciales de Programación. Un curso habitual consta de clases teóricas, en su mayoría expositivas por parte de los docentes, y de clases prácticas en los laboratorios. Aquí se presenta la incorporación sistemática a las clases teóricas de las KLA ("Kinesthetic Learning Activities").

Primeramente se describen varios aspectos de cursos de Programación I y se describen qué son las KLA. Después se presenta el detalle del curso de Programación I en la Universidad ORT y la incorporación de las KLA. Se ofrece la experimentación y el análisis de los resultados finales de los trabajos de los alumnos y de sus percepciones obtenidas a partir de una encuesta. Por último, se brindan conclusiones y propuestas de trabajo futuro.

2 Cursos de Programación I y KLA

El aprendizaje es un proceso activo, no pasivo. Para lograr que los alumnos alcancen niveles óptimos de competencia, deben ser motivados a involucrarse de

forma activa en el proceso de aprendizaje, en actividades que incluyan por ejemplo, resolver problemas reales, producir trabajos escritos originales, realizar proyectos de investigación científica, construir objetos y dialogar con otros acerca de temas importantes [1].

Programar es una actividad compleja. Es importante tener en cuenta que el curso de Programación I esté bien construido [2]. Robins, Rountree y Rountree [3] indican que aprender a programar implica adquirir un conocimiento completamente nuevo y una serie de estrategias y habilidades prácticas. Gries [4] señala que es necesario revisar seriamente cómo se enseña a programar. El propósito de la educación no debe ser solamente exponer hechos a los estudiantes, sino enseñarles a pensar. En [5], en particular refiere a varios principios aplicables a la enseñanza, entre ellos presentar los conceptos con el nivel apropiado de abstracción.

El análisis de los cursos de Programación I puede hacerse desde diferentes enfoques, no necesariamente disjuntos. Estos podrían incluir, entre otros, la orientación principal del curso, las herramientas a utilizar y las metodologías. En relación a la orientación principal del curso podrían considerarse por ejemplo, la prueba de programas -“test driven development”- [6] [7]. [8], la ingeniería de software [9] o la programación en pares -pair-programming- [10]. En cuanto a qué herramientas utilizar en el curso, Powers y colegas [11] las clasifican en varias categorías, algunas de ellas son: a) herramientas narrativas que permiten programar para contar una historia (son motivadoras), b) herramientas visuales que permiten construir programas a través de arrastrar elementos (ofrecen flexibilidad), y c) herramientas basadas en el modelado del flujo que construyen el programa conectando elementos para representar el orden del mismo (reducen la complejidad). Finalmente, en relación a metodologías concretas, Dugas [12] señala el uso de un “quiz” diario como una técnica de aprendizaje activo y cooperativo. Cada quiz tiene como función reflejar los objetivos de la clase y servir como guía para el desarrollo de otros materiales. Su estudio muestra que tuvieron mejores resultados los alumnos que participaron en ese curso. Sung y colegas [13] promueven la integración de la programación de juegos y gráficos.

Otra herramienta vinculada a la metodología y con énfasis en el aprendizaje activo son las KLA. Una KLA es una herramienta pedagógica que involucra movimientos físicos de los estudiantes [14]. Las características básicas de este tipo de actividad son: los estudiantes están activa y físicamente participando en la exposición y asimilación del material de clase y, esta participación está relacionada directamente a objetivos de aprendizaje específicos. Señala que las KLA promueven la interactividad entre los estudiantes y mejora el aprendizaje. Formula varias recomendaciones para el diseño, la elaboración, la adaptación y la ejecución de las diferentes KLA, como por ejemplo incluir elementos sorpresa, involucrar a toda la clase y dar instrucciones simples. Begel y colegas [15] señalan que en general las KLA son breves (20 minutos o menos) y que pueden involucrar a todos o algunos de los estudiantes. Pueden ser utilizadas en grupos grandes o pequeños. Astrachan [16] presenta varios ejemplos que son en cierta forma casos de KLA. Entre ellos ofrece uno para representar el pasaje de parámetros usando discos voladores. Destacan que el uso de este tipo de ejercicios

proveen una imagen física que ayuda a los estudiantes a construir modelos mentales. Anewalt [17], en su curso incluye varias KLAS y refiere a que ello fomenta y logra que los estudiantes vengan a clase “despiertos y curiosos” sobre qué ocurrirá cada día, con alto grado de motivación.

Del relevamiento bibliográfico se constata la oportunidad de colaborar en la búsqueda de elementos que promuevan la comprensión, la abstracción, el desarrollo de capacidades así como el desarrollo de estrategias de resolución de problemas para el curso de Programación I. En el marco del aprendizaje activo, las KLA pueden ser una herramienta de utilidad.

3 Programación I en Universidad ORT Uruguay

El curso de Programación I en Universidad ORT Uruguay tiene por objetivo iniciar la enseñanza de la programación empleando técnicas de orientación a objetos. Sus contenidos incluyen conceptos básicos de Programación, conceptos básicos de orientación a objetos y su aplicación en un lenguaje de programación, en particular Java. 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.

El curso consta de 15 semanas, cada una con 4 horas de clase teórica y 2 horas de práctica en laboratorio. El detalle del curso se ofrece en la Tabla 1.

Tabla 1. Descripción del curso de Programación I

Semanas	Contenidos
1-3	Estructuras de control, pseudocódigo y variables
4	Clases y objetos, clases standard
5-8	Creación de nuevas clases, manejo de relaciones y alias
9	Herencia
10-12	Colecciones, ordenación y búsqueda
13	Excepciones
14-15	Manejo avanzado de Colecciones

Para aprobar el curso se deben realizar satisfactoriamente 3 tareas: 2 trabajos obligatorios en máquina (en grupos de 2 personas) y un parcial individual. El primer obligatorio implica generalmente el desarrollo de un pequeño sistema (3-4 clases) y el segundo es una ampliación y modificación del primero. En este caso, se agregan otras clases y procesos varios, con algoritmia interesante. Por cada trabajo hay una defensa individual. El primer trabajo vale 25 puntos, el segundo 30 y el parcial 45. En caso de obtenerse 86 puntos o más, además se exonera la materia y no debe rendirse examen. Si se obtiene entre 70 y 85 puntos se aprueba el curso y debe rendirse examen. Con menos de 70 puntos se pierde el curso, lo que impide continuar con el curso de la materia siguiente (Programación II).

En el curso se utilizaron 10 KLA. Se usó una por semana. Cada KLA lleva entre 15 y 20 minutos. En la selección y el diseño de las KLA para este curso se tuvieron en cuenta las recomendaciones presentadas [14], [15] y también se hizo énfasis en considerar los distintos estilos de aprendizaje [18]. Se presenta en la Tabla 2 el título y semana de uso de cada una de ellas.

Tabla 2. Resumen de KLA

Semana	KLA	Semana	KLA
1	1- Dibujos	7	6 - Isomorfismo
2	2- Helado	8	7 – Cambios en el proceso
3	3- Shuttle Puzzle	11	8 – Ordenación
5	4- Masa de modelar	14	9 – Algoritmia
6	5- Juego de rol	15	10 – Jeopardy

3.1 KLA 1 - Dibujos

La KLA 1 incluye un conjunto de láminas grandes con dibujos. Cada lámina debe ser descrita oralmente por un estudiante a otro, quien debe hacer el dibujo en el pizarrón sin ver el original. Las láminas incluyen, por ejemplo, un cuadrado y un triángulo dispuestos uno sobre otro o círculos concéntricos. El estudiante que dibuja simula la computadora y el que describe indica la instrucción a ejecutar. Trata de mostrar la necesidad de ser preciso al momento de dar instrucciones, o sea, al elaborar un algoritmo.

3.2 KLA 2 - Helado

Para el desarrollo de sistemas grandes se hace necesario el trabajo en equipo. Se fomenta esta habilidad desde el comienzo mismo de la carrera, pues los trabajos obligatorios son en grupo. Como indica Goleman [19], la excelencia en estos tiempos depende más de las competencias emocionales que de las capacidades cognitivas. Indica de habilidades sociales (capacidad para inducir respuestas deseables en los demás) y habilidades de equipo (para crear la sinergia grupal en la consecución de metas colectivas). En este sentido, el ejercicio que propone y se adapta aquí como KLA es el del "Helado".

Esta KLA pide "indicar cuál es el sabor de helado más rico del mundo". La forma de plantear este ejercicio es: a) indicar las reglas del trabajo, y b) formular en el pizarrón la pregunta. Las reglas son informar que se realizará una pregunta, de la cual no se dará más información adicional y que cada equipo de 4 ó 5 alumnos dispondrá de 10 minutos para dar una única respuesta. Es recomendable escuchar los debates que se generan en cada grupo y tomar nota de los comentarios, pues serán la base de la discusión posterior. Al cabo del tiempo establecido cada grupo enuncia su respuesta. Eventualmente algunos podrán indicar que no llegaron a un acuerdo. Como cierre del trabajo, se describen cada uno de los comentarios anotados y se discuten, destacando fortalezas y debilidades. Algunos ejemplos son: "es necesario establecer

un criterio para saber el mejor sabor" (refiere a establecer una forma de trabajo); "votemos el mejor sabor" (solución por consenso), "partamos el grupo y demos 4 respuestas" (se contradice con la premisa de respuesta única, genera conflicto), "elijamos sabor crema que es el que me gusta a mí" (falta de tolerancia), "¿Chocolate? ¿Quién dijo que es rico?" (actitud que genera reacciones negativas), y "no me dejaron decir mi sabor preferido" (falta de participación de un integrante). También se vinculan estas actitudes a la resolución de problemas en equipo.

3.3 KLA 3 – Shuttle Puzzle

Tiene por objetivo presentar la idea que para desarrollar un algoritmo que resuelva un problema se debe entender claramente el mismo. Hay múltiples formas para resolver problemas. Entre los métodos probablemente más conocidos se puede indicar el de Descartes [20] y Polya [21]. Gray [22] señala que la resolución exitosa de problemas requiere habilidades de alto nivel de pensamiento. Ejemplificando con el puzzle "Shuttle" da una lista de recomendaciones para el proceso de resolver problemas. El puzzle consiste en 4 bolitas o palitos de un color, 4 de otro color diferente y un bloque de madera con 9 agujeros alineados; los palitos del mismo color están puestos en los extremos, dejando el centro vacío. El objetivo es invertir los palitos, siguiendo las reglas de: a) avanzar una posición un palito a un lugar vacío adyacente, y b) saltar una posición sobre un palito del color contrario a una posición vacía.

Se reparte un juego de palitos a cada equipos de 3 o 4 estudiantes, se detallan las reglas y se les pide que lo resuelva. Este proceso es observado y se toma nota de qué es lo hacen los estudiantes. Se observa así que algunos pocos estudiantes lo resuelven, otros detectan que no entendieron las reglas y abandonan, y algunos han generado ideas para nuevos problemas a partir del original. Así, algunas de las estrategias utilizadas comúnmente por los alumnos son: prueba y error, simplificación del problema considerando 2 ó 3 palitos en vez de 4; e intención de deducir reglas. Como conclusión del juego, se discuten las observaciones realizadas y se anota en el pizarrón las recomendaciones hacia la resolución de problemas, que combinan y resumen los métodos generales presentados: a) entender el objetivo, b) entender las reglas, c) evaluar alternativas (simplificar, ensayo y error, analogías, etc.), d) resolver el problema, y e) verificar.

3.4 KLA 4 – Masa de modelar

Está enfocada en favorecer la comprensión de la diferencia entre objeto y clase, y familiarizarse con los conceptos de igualdad, alias de objetos y pasaje de mensajes. Se reparten potes de masa de modelar y moldes, y se crean diversos objetos [17]. Los moldes representan las clases. Los elementos cortados son los objetos. Se amplía la propuesta de [17] representando la asociación de una variable (definida en Java) anotada en el pizarrón con un objeto a través de un hilo que va adherido desde el pizarrón al propio objeto creado en masa. Se muestra especialmente la creación,

asignación e identidad de objetos y el envío de mensajes. Asimismo, se muestra el concepto de alias, cuando otra variable “mira” al mismo objeto (representado por otro “hilo” desde el pizarrón al objeto de masa).

3.5 KLA 5 – Juego de Rol

Esta herramienta es una adaptación de la presentada en [23]. El objetivo es presentar los conceptos básicos de objetos con una aplicación real. En particular, se trata de comprender mensajes y métodos. Las clases definidas son Calculadora común (que sabe sumar y restar), Calculadora Creativa (que suma y resta, pero usa un método especial para restar), Calculadora Burocrática (que utiliza en sus métodos a calculadoras comunes), Acróbata (que saben aplaudir, sentarse y pararse) y Dado. Se pega en las paredes carteles con la lista de mensajes que entiende cada clase, se asignan roles a los estudiantes y se les da mensajes. Cada estudiante debe representar físicamente la ejecución. Mensaje, método, polimorfismo, pasaje de parámetros, diferencia entre clase e instancia, variables privadas, son algunos de los conceptos que se presentan en el ejemplo.

3.6 KLA 6 – Isomorfismo

El concepto matemático de morfismo establece una correspondencia entre diferentes dominios, mientras se mantienen las propiedades. Un modelo es isomorfo respecto al sistema que trata de reproducir cuando el dominio original y el modelo pueden ser considerados equivalentes. Cuando una situación se modela con un morfismo, es porque hay cosas o casos que son irrelevantes en el dominio original. En un sentido amplio y con enfoque informático, en vez de considerar que en los morfismos se conservan las operaciones, se podrían considerar que conservan los comportamientos. Todas estas definiciones son presentadas en la clase.

La KLA 6 plantea resolver un problema isomorfo al TaTeTi. Se les entrega un plano a cada par de estudiantes y deben colorearlo según ciertas restricciones. Se trata de establecer isomorfismos como herramienta de resolución de problemas. En particular, se propone el ejercicio "JAM" [24]. En este ejercicio, hay rutas numeradas que van entre diferentes ciudades. El objetivo del juego es pintar con el mismo color tres caminos que lleguen a una misma ciudad. Cada jugador tiene un color. En su turno respectivo, el jugador pinta una carretera en su totalidad. Este ejercicio es isomorfo al conocido TaTeTi. Algunos estudiantes indicaron que "es similar al TaTeTi"; "perder en este juego depende de que un jugador se distraiga, es análogo al TaTeTi"; otro indicó: "es un juego de estrategia, hay que tratar de anticipar la jugada del oponente". Se observa la aplicación de los conceptos de morfismos al análisis de este problema particular.

3.7 KLA 7 – Cambios en el proceso

Está orientada a gestionar los cambios de requisitos durante el desarrollo. Los estudiantes manejan *puzzles* de goma con los que deben formar ciertas figuras y luego se cambian los requerimientos.

Este ejercicio promueve la evaluación de alternativas y la necesidad de considerar "rediseñar" una solución, como una estrategia más de resolución de problemas. En el ejercicio propuesto por De Bono [25], se solicita ir construyendo figuras geométricas reconocibles a partir de ciertas fichas. Se entrega a cada alumno un conjunto de fichas numeradas y se le indica que arme una figura geométrica reconocible utilizando dos piezas determinadas. Luego se señala que se incorpore otra ficha dada, formando con las 3 piezas una figura reconocible. Posteriormente se indica agregar otras dos piezas determinadas. El último paso es incorporar la última ficha. En la gran mayoría de los casos, la figura armada no permitirá agregar esta última ficha. Así, si se empieza con la construcción de, por ejemplo, un cuadrado, no es posible incorporar la última pieza, es necesario volver atrás y reestructurar el modelo. A pesar de haber procedido correctamente en cada fase, se requiere reestructurarlo para poder incluir la nueva pieza. Como culminación del trabajo, se reflexiona además sobre las formas de resolver problemas.

3.8 KLA 8 – Ordenación

Para presentar los conceptos de ordenación, se realiza esta KLA que consiste en un conjunto de revistas que se distribuyen a los estudiantes y ellos deben manipular y ordenar. En primera instancia, se les solicita a algunos que ordenen las revistas según un criterio definido y se hace énfasis en descubrir posibles algoritmos de ordenación (selección, intercambio, enumeración, etc.). En una etapa posterior, con la intención de presentar la idea de que es posible tener un algoritmo que ordene elementos en general y se puede "separar" el criterio de ordenación (en Java sería el uso del método *Collections.sort()* y la *interface Comparable*), se le asigna una revista a cada estudiante y en secreto se les indica cuál es el criterio. El estudiante que simula el algoritmo de ordenación, debe ir consultando de a pares, según el algoritmo que se siga, si una revista va antes o no que otra. Se puede extender también para incluir los conceptos de *Comparator*.

3.9 KLA 9 – Algoritmia

En esta KLA, basada en [26], se dispone en el pizarrón una grilla rectangular de cartas, algunas de ellas con su dorso al frente y otras al revés. Inicialmente se le pide a un alumno que las pegue según su preferencia, luego, con la "excusa" de que el tamaño es chico, se agregan más cartas. Estas están dispuestas de forma que haya paridad par de cartas con dorso al frente tanto en forma horizontal como vertical. Se le pide a otro alumno que, sin mirar el docente, de vuelta alguna carta. "Mágicamente" el profesor puede detectar cuál fue modificada. Este ejercicio permite

presentar el concepto de paridad, y se aplica a algoritmos de recorrida de *arrays* para chequear si la paridad de un *array* binario es impar. En su primera versión, la gran mayoría de los estudiantes utilizan un contador para llevar la cantidad de posiciones en 1, y al final chequean si es cantidad impar o no. El nuevo requerimiento es que hagan el mismo algoritmo pero sin usar el contador. Esto permite reforzar la idea de paridad, pues basta una variable booleana que vaya alternando su valor con cada nuevo “1” para realizar este chequeo. Fomenta asimismo la confección de diversos algoritmos para un mismo problema.

3.10 KLA 10 - Jeopardy

“Jeopardy” es un juego en el cual hay varios temas, cada uno con 5 o 6 preguntas, de valores crecientes [27]. En este curso, algunos de los temas elegidos fueron: Java, Siglas, Lenguajes, y Funcionamiento de la universidad. La implementación puede ser con cartones cada uno con una pregunta o utilizar un programa que realice la gestión. Se arman 2, 3 ó 4 equipos de estudiantes (dependiendo el tamaño de la clase). En su turno, pasa un estudiante del equipo e indica qué tema y cuál pregunta. En caso de responder correctamente, su equipo suma el puntaje respectivo. Este ejercicio tiene por finalidad repasar conceptos. Da lugar a una interesante discusión pues hay varias preguntas que pueden originar dudas [15].

4 Experimentación y resultados obtenidos

La experimentación se realizó en el segundo semestre de 2008. Se seleccionaron aleatoriamente 3 grupos: 1 de prueba y 2 de control. Los alumnos se asignaron al azar a cada uno de los grupos. Cada grupo tiene entre 20 y 22 alumnos. Para la evaluación de la experimentación se tomaron medidas genéricas: resultado de curso, en cuanto a eliminación, aprobación y exoneración. Es de destacar que para la corrección de toda tarea se aplican pautas elaboradas en conjunto por la Cátedra. Además, se realiza doble chequeo, esto es, dos docentes revisan cada trabajo independientemente. Todos los docentes de estos 3 grupos cuentan con varios años de experiencia. El material teórico y práctico [28] es común a todos los estudiantes. Los resultados se presentan en la siguiente tabla (Tabla 3).

Tabla 3. Resultados del curso de Programación I

Grupo	Grupo Control I	Grupo Control II	Grupo Experimental
Inscriptos (IN)	22	20	22
Aprobados (AP)	18	13	17
Eliminados (EL)	4	7	5
Exonerados (EX)	6	3	14
% Aprobación (AP*100/IN)	82%	65%	77%
% Exoneración (EX*100/IN)	27%	15%	64%
% Eliminación (EL*100/IN)	18%	35%	23%

Los valores de aprobación son relativamente similares en los tres grupos, entre 65-82 %. Se observa un valor notoriamente más alto de exoneración (64%) en el grupo de Experimentación en relación a 27% y 15% en los de control.

Las encuestas realizadas a los alumnos del Grupo Experimental incluían preguntas acerca de qué utilidad le encontraron a las actividades realizadas. Se les preguntó en particular sobre algunas específicamente y también en términos generales. De la KLA 1, señalaron: “me pareció una dinámica interesante que me ayudó a comprender”, “muy bueno. Parecía muy simple a primera vista pero es complejo llevarlo a la práctica”, “permite crear un paralelismo entre las diversas maneras de razonar”.

En relación al uso de masa de modelar (KL4) indicaron: “buen apoyo grafico para aclarar la creación de instancias”, “completamente útil”, “super didáctico”, “comprensible”, “no lo entendí en el momento, le encontré más coherencia con el paso del tiempo”. El juego de rol (KL5), según expresaron, les ayudó a comprender claramente los conceptos y a entender el envío de mensajes y pasaje de parámetros.

Del Jeopardy (KLA 10) expresaron: “interesante y muy divertido”, “sirve para aprender cositas que hay que saber”, “promueve el trabajo en equipo y repasa conceptos”, “no sabía muchas de las respuestas, pero fue una instancia más de aprendizaje”, “ayuda a reafirmar conceptos “jugando””. En relación a la impresión general y utilidad sobre las actividades indicaron aspectos como “excelente pedagogía”, “actividades esclarecedoras y estimulantes” y “una buena forma de entender”

5 Conclusiones y trabajo futuro

De acuerdo a los resultados de la experimentación, como se presentó, se obtienen valores destacablemente más altos en cuanto a exoneración de la materia. Se observa que los estudiantes que participaron en este curso obtienen significativamente mejores resultados que en los cursos tradicionales. Además, sus comentarios en la encuesta destacan un alto grado de motivación e interés. Puede concluirse que el uso sistemático de KLA permite a los alumnos obtener mejores resultados. Como trabajo futuro, se propone continuar en la búsqueda de elementos que permitan generar nuevas KLA y, o, adaptar las disponibles.

Referencias

1. Khvilon, E. (coordinador), Las tecnologías de la información y la comunicación en la formación docente. Guía de planificación, UNESCO. Gráfica Futura, Uruguay (2004).
2. Pendergast, M.: Teaching Introductory Programming to IS Students: Java Problems and Pitfalls, Journal of Information Technology Education, Vol 5, 491-515 (2006)
3. Robins, A., Rountree, J. and Rountree, N.: Learning and teaching programming: a review and discussion, Computer Science Education, Vol 13, No 2, 137-172, USA (2003)

4. Gries, D.: What have we not learned about teaching Programming. Computer, pp 81-82, October (2006)
5. Gries, D.: A Principled approach to teaching OO First. En: Proc. of SIGCSE 2008, pp 31-35, USA (2008)
6. Janzen, D., Saiedian, H.: Test-Driven Learning in Early Programming Courses. En: Proc. of SIGCSE 2008, USA (2008)
7. Girard, D., Wellington, C.: Work in Progress: A test-first Approach to teaching CS1. En: Proc. of 36th ASEE/IEEE FIE, M3E-19, USA (2006)
8. Briggs, T., Girard, D.: Tools and techniques for test-driven learning in CS1. En: JCSC 22, 3. (2007)
9. Rao, T., Mitra, S.: An Early Software Engineering Approach to Teachint CS1, CS2 and AI. En: Proc. of SIGCSE 2008, USA (2008)
10. Hanks, B.: Student performance in CS1 with distributed pair programming. En: Proc. of ITICSE 2006, Portugal (2006)
11. Powers, K., Gross, P., Cooper, S., Goldman, K., Carlisle, M., Mc Nallyu, MI, Prouls, V.: Tools for teaching introductory programming: What works?. En: Proc. of SIGCSE'06, USA (2006)
12. Dugas, C.: No Computers? No Problem! Active and cooperative learning in an introductory computer science course. En: Proc of 38 th ASEE/IEEE FIE Conf., USA (2008)
13. Sung, K., Panitz, M., Rosenberg, R., Anderson, R.: Assessing Game-themed programming assignments for CS1/2 Courses. En: Proc. of GDCSE'08, USA (2008)
14. Sivilotti, P, Pike, S.: A collection of kinesthetic learning activities for a course on distributed computing. En: ACM SIGACT news, V. 38, Issue 2, pp. 56-74. USA (2007)
15. Begel, A., Garcia, D., Wolfman, S.: Kinesthetic Learning in the Classroom. En: Proc. of SIGCSE 2004, pp 183-184 USA (2004)
16. Astrachan, O.: Concrete teaching: hooks and props as instructional technology. En: Proc. of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education, Ireland. (1998)
17. Anewalt, K.: Making CS0 Fun: an active learning approach using toys, games and Alice. En: Consortium for Computing Sciences in Colleges Eastern Conference, USA (2007)
18. Felder, R, Silverman, L.: Learning and Teaching Styles in Engineering Education. En: Engr. Education, V. 78 No. 7, pp 674-681 (1988)
19. Goleman, D. : La práctica de la inteligencia emocional. Ed Kairós, España (1999)
20. Descartes, R.: *Discurso del método*. 5ta edición. Editorial Sopena Argentina, Argentina (1964) (edición original 1637).
21. Polya, G.: How to solve it. A new aspect of mathematical method. Princeton University Press (USA) (1988) (1era. edición: 1945).
22. Gray, J.S.: The Shuttle puzzle: a lesson in problem solving. Journal of computing in higher education, V. 10, No. 1, pp 56-70 (Fall 1998)
23. Bergin, J. ,<http://csis.pace.edu/~bergin/Java/RolePlay.html>
24. <http://www.juntadeandalucia.es/averroes/iesarroyo/matematicas/taller/aspectosweb/aspectosweb.htm>
25. de Bono, E.: El pensamiento lateral. Ed. Paidós, Barcelona (1998)
26. Computer Science Unplugged, <http://csunplugged.org/index.php/en/04-error-detection-activitiesmenu-110>
27. Bergin, J.: Academic Jeopardy, En: Proc. of the 10th annual SIGCSE, Portugal (2005)
28. Kerekí, I.: Programación I: Aprender programación orientada a objetos desde cero. Empresa Gráfica Mosca, Uruguay (2008)