

Aprendizaje colaborativo con técnicas de realidad aumentada como estrategia didáctica para la enseñanza de estructuras iterativas en fundamentos de programación de computadores

Javier A. Jiménez Toledo
Grupo Tecnofilia
I.U. CESMAG
jajimenez@iucsmag.edu.co

Cesar A. Collazos
Grupo IDIS
Universidad del Cauca
ccollazo@unicauca.edu.co

Julio A. Hurtado
Grupo IDIS
Universidad del Cauca
ahurtado@unicauca.edu.co

Libardo Pantoja Yépez
Grupo IDIS
Universidad del Cauca
wpantoja@unicauca.edu.co

Fecha Recepción: 24/10/14 - Fecha Aprobación: 4/12/14

Resumen: Este artículo, resultado de investigación, presenta la experiencia de validación de un modelo de aprendizaje colaborativo apoyado con técnicas de realidad aumentada como estrategia didáctica para la enseñanza de las estructuras iterativas, con estudiantes de primeros cursos de fundamentos de programación en instituciones de educación superior de modalidad presencial. El diseño de investigación planteado evaluó el tratamiento experimental, basado en aprendizaje colaborativo, cuyos resultados generaron interesantes indicadores de rendimiento académico con altos niveles de expectativas de aprendizaje en los estudiantes.

Palabras clave: Aprendizaje colaborativo, fundamentos de programación de computadores, realidad aumentada.

Abstract: This paper presents the experience of validating a model supported collaborative learning with augmented reality techniques as a teaching strategy for teaching students the cyclic structures of first programming fundamentals courses in institutions of higher education modality. The proposed research design evaluated the experimental treatment based on collaborative learning, the results generated interesting indicators of academic achievement with high expectations for student learning.

Keywords: Collaborative learning, fundamentals of computer programming, augmented reality, cycles.

Introducción

La programación de computadores es un área de conocimiento donde confluyen conceptos y habilidades esenciales para la práctica de la programación independiente del paradigma (Association for Computing Machinery, 2009). Pese al estado tecnológico actual y la utilización del software en todos los campos de actuación humana, existen algunos problemas relacionados con la fundamentación de los futuros constructores de software, debido a una cantidad de situaciones que van desde la carencia de habilidades para programar, el desconocimiento de la materia y hasta la carencia de disciplina en programación (Oviedo, 2002). Dichos inconvenientes se gestan desde el primer curso de programación recibido, el cual es clave en el proceso de

formación que tendrá un desarrollador de soluciones software en su etapa de aprendizaje y más tarde en su vida profesional (Hernández, Jiménez, & Martínez, 2012).

La enseñanza de los fundamentos de programación ha llevado a muchas reflexiones por las etapas que de acuerdo a la evolución de sus propios paradigmas se han generado. Establecer relaciones entre la evolución del pensamiento y los paradigmas de programación permite encontrar algunas relaciones causales entre cada uno de los paradigmas y la forma de modelar y enseñar soluciones computacionales (Buritica & Ivan, 2012). Es así como en los primeros cursos algunos docentes inician la enseñanza de los fundamentos de programación, independiente del paradigma seleccionado (procedural, orientado a objetos,

funcional o lógico), con contextos relacionados al campo de la matemática y la geometría, ya que los ejemplos matemáticos básicos y, especialmente, los de geometría plana elemental son posibles de modelar debido a que los estudiantes han adquirido suficientes conocimientos básicos previos en esas disciplinas (Bett et al., 2013). En cambio, otros docentes llevan al estudiante a plantear soluciones a problemas a través de un proceso cíclico de "prueba y error" hasta obtener una solución que satisfaga las normas de evaluación propuestas (Murillo Rivera, 2006). Además, dentro de un paradigma determinado se visualizan varios enfoques para enseñar: algunos enseñan a programar en un lenguaje de programación particular, utilizando su sintaxis y su semántica, y otros emplean un lenguaje algorítmico lo bastante general como para permitir su traducción posterior a cualquier lenguaje de programación (Szpiniak & Rojo, 2007).

En la enseñanza de los fundamentos de programación las estructuras iterativas tienen especial tratamiento, puesto que en principio su apropiación, por parte de los estudiantes, resulta compleja y para la gran mayoría (que por primera vez se enfrentan a su estudio) se dificulta su comprensión por la falta de asociación con sus experiencias vivenciales.

Por lo anterior, en este artículo se presenta una alternativa diferente para acercar al estudiante de programación computacional, de una manera práctica, a tener una visión general del comportamiento de una estructura iterativa, mediante la integración de ejemplos construidos con técnicas de realidad aumentada y soportados con una estrategia didáctica colaborativa de aprendizaje.

Dificultades en la enseñanza de los fundamentos de programación de computadores.

Aprender a programar es reconocido por ser difícil, una de las mayores dificultades es que en la programación los lenguajes son artificiales (Mosser, 1997). Truong, Bancroft y Roe establecen que la educación en computación se enfrenta a tres grandes problemas: en primer lugar, la mayoría de los novatos estudiantes de programación tienen dificultad al enfrentarse a su primer programa o en la construcción de conocimientos referentes a programación. En segundo lugar, la gran cantidad de estudiantes matriculados en los cursos de introducción a la programación provoca

dificultad en propiciar un ambiente de aprendizaje efectivo. Por último, hay una creciente necesidad de apoyar los campus desde múltiples sitios que redundan en el manejo de grandes cantidades de información técnica al estudiante (Truong, Bancroft, & Roe, 2003).

En el estudio realizado por Affleck y Smith se ha encontrado que el principal problema de los programadores principiantes es el acceso a los conocimientos previos y la adopción de un enfoque para estudiar, que va más allá de la memorización explícita de conocimientos necesarios para aplicar y transferir el dominio de conceptos a situaciones nuevas (Affleck, G. and Smith, 1999). Es así como los métodos de representación de lógica formal como pseudocódigo, diagramas de flujo de datos, diagramas Nassi-Shneiderman y en otros casos el manejo de paradigmas como el Orientado a Objetos, Lógico o Funcional, que se imparten en los primeros cursos de programación, no vivencian explícitamente los resultados de operación y ejecución de una manera clara en sus fases iniciales. Esto conlleva a que el estudiante necesariamente tenga que regirse por las recomendaciones del docente frente a una solución planteada en un caso de estudio, hasta que su veracidad sea demostrada mediante una herramienta (posiblemente de software), que en muchos casos no es utilizada como mecanismo de soporte al proceso lógico llevado a efecto en los inicios de tales cursos.

Para Manuel Gonzales y Maximiliano Paredes son muchos los inconvenientes que debe enfrentar un estudiante de primer curso de programación debido a la complejidad de las temáticas tratadas. Por ello, el inicio en el aprendizaje de la programación siempre ha sido un proceso complicado, ya que es una disciplina totalmente diferente a lo que se ha visto desde entonces por parte del programador novato, porque la programación exige cambiar de manera radical el modo de pensar y analizar las cosas y aun habiendo adquirido los conocimientos teóricos necesarios se ha detectado una gran dificultad de aplicar esos conocimientos teóricos en la resolución de problemas prácticos (González De Rivera Fuente & Paredes Velasco, 2008). Además, en el momento de presentar al estudiante conceptos referentes a procesos de entrada, salida, estructuras condicionales y estructuras iterativas (que en la mayoría de los casos no ha tenido experiencia alguna en el campo de la programación), éste tiene problemas en la asimilación y por ende en

el modelamiento de tales concepciones, puesto que la abstracción de estos modelos forman parte del desarrollo de pensamiento complejo. La enseñanza de la programación no se puede transmitir directamente desde instructores a los alumnos, debe ser adquirida activamente por los estudiantes (Ben-Ari, 2001).

Metodologías para el aprendizaje de programación de computadores.

Es evidente la preocupación existente en los docentes de los primeros cursos de programación con relación a los resultados obtenidos en el proceso de aprendizaje, por ello se han llevado a cabo en los últimos años varios proyectos encaminados a mejorar dichos procesos, donde la mayor atención se enfoca en los primeros niveles de formación, por ello existen diversas metodologías que sirven como herramientas de apoyo en la consolidación de las competencias necesarias que un desarrollador de software debe tener en su formación. En la tabla 1 se categorizan cada una de las metodologías mencionadas de acuerdo a sus características afines.

Tabla 1. Metodologías para la enseñanza de los fundamentos de programación de computadores.

Metodología	Características integradas
EDCIA Herramientas y técnicas para para enseñanza de la programación PLDETECTIVE BlueJ	Generación de código a lenguajes de programación, editor gráfico y ofrece software de modelado.
PBL y CSCL COLLEGE OOP ANIM HABIPRO JECO DPE	Chat, edición compartida, síncrono y ofrece software de modelado.
PBL y CSCL Cupi 2	Chat, edición compartida, conferencia, síncrono, asíncrono y utiliza lenguajes de programación existentes.
ALGOARENA	

Metodología	Características integradas
OOP ANIM HABIPRO JECO ProLearn DPE VPL MILLENIUM Cupi2	Roles de docente y estudiante
ALGOARENA ProLearn PLDETECTIVE VPL MILLENIUM Cupi2	Metodología de auto aprendizaje.
EDCIA PBL y CSCL ALICE GREENFOOT SCRATCH SIGACLE ALGOARENA COLLEGE OOP ANIM HABIPRO JECO DPE ELP VPL MILLENIUM Cupi2	Síncronas
PBL y SCCL ALICE GREENSOFT DPE PLDETECTIVE BlueJ VPL MILENIUM Cupi2	Asíncronas
COLLEGE OOP ANIM	

Metodología	Características integradas
HABIPRO JECO DPE ELP BlueJ VPL MILLENIUM	Aprendizaje basado en colaboración

Aprendizaje colaborativo y estrategias didácticas

El aprendizaje colaborativo (cooperativo) es el uso instruccional de pequeños grupos de tal forma que los estudiantes trabajen juntos para maximizar su propio aprendizaje y el de los demás (Jonhson, Jonhson, & Holubec, 1993), donde cada miembro del grupo de trabajo es responsable no solo de su aprendizaje sino de ayudar a sus compañeros a aprender, creando con ello una atmósfera de logro (Monterrey, 2008). Los estudiantes trabajan colaborando, este tipo de aprendizaje no se opone al trabajo individual ya que puede observarse como una estrategia de aprendizaje complementaria que fortalece el desarrollo global del alumno. Además, se establece que los métodos de aprendizaje colaborativos traen consigo una renovación en los roles asociados a profesores y alumnos, en el caso de los profesores se establece tres tipos: Profesores como Mediador cognitivo, Instructor y Diseñador Instruccional (C. Collazos, Guerrero, & Vergara, 2012).

En su sentido básico, aprendizaje colaborativo se refiere a la actividad de pequeños grupos desarrollada en el salón de clase (Monterrey, 2004), de tal manera que los alumnos forman "pequeños equipos" después de haber recibido instrucciones del profesor y dentro de cada equipo los estudiantes intercambian información y trabajan en una tarea hasta que todos sus miembros la han entendido y terminado, aprendiendo a través de la colaboración. Comparando los resultados de esta forma de trabajo, con modelos de aprendizaje tradicionales, se ha encontrado que los estudiantes aprenden más cuando utilizan el aprendizaje colaborativo, recuerdan por más tiempo el contenido, desarrollan habilidades de razonamiento superior y de pensamiento crítico y se sienten más confiados y aceptados por ellos mismos y por los demás (Millis, 1996).

La implementación de actividades colaborativas, con el fin de mejorar el proceso en enseñanza aprendizaje en el aula, incluye básicamente dos acciones de gran importancia: una primera etapa de configuración de las condiciones iniciales y una segunda que garantice la aplicación de los principios básicos de una práctica colaborativa (C. A. Collazos & Mendoza, 2006). En tal sentido, en el rol de profesor como Diseñador Instruccional, el docente se encarga de definir las condiciones iniciales del trabajo, planteando los

objetivos académicos, definiendo claramente las unidades temáticas y los conocimientos mínimos que deben ser adquiridos durante el proceso de enseñanza en cada una de ellas. Esto requiere adicionalmente, explicar los criterios de éxito, definir las tareas a realizar con unos objetivos claramente definidos, explicar claramente los conceptos que subyacen el conocimiento de cada temática, definir los mecanismos de evaluación que se tendrán y monitorear el aprendizaje de los alumnos dentro de la sala de clase (C. Collazos et al., 2012).

A su vez, la didáctica se define como la técnica que se emplea para manejar, de la manera más eficiente y sistemática, el proceso de enseñanza-aprendizaje (De la Torre Zermeño, 2005). Las estrategias didácticas contemplan las estrategias de aprendizaje y las estrategias de enseñanza, donde las primeras consisten en un procedimiento o conjunto de pasos o habilidades que un estudiante adquiere y emplea de forma intencional, como instrumento flexible para aprender significativamente y solucionar problemas y demandas académicas. Por su parte, las estrategias de enseñanza son todas aquellas ayudas planteadas por el docente, que se proporcionan al estudiante para facilitar un procesamiento más profundo de la información (Díaz Barriga & Hernández Rojas, 1999).

Los distintos cambios experimentados por las sociedades han planteado necesidades formativas que requieren un estudio pormenorizado de las estrategias de formación, cualificación y acreditación por parte de las instituciones universitarias. En ese sentido, las fórmulas docentes deben pensarse desde espacios educativos que fomenten el aprendizaje autónomo e integren conocimientos y competencias universitarias a la vez. Se trata de redimensionar y contextualizar principios generales de la enseñanza y el aprendizaje con estrategias didácticas e incorporarlas a la programación de contenidos y a los métodos de aprendizajes, a los procesos de flexibilidad mental en los espacios complementarios de clase, entendidos como trabajo de asesoramiento docente de orden formativo, científico y profesional que permita otras prácticas de enseñanza docente, y la utilización del Aprendizaje Colaborativo (Fonseca & Aguaded Gómez, 2007).

Realidad aumentada RA

A pesar de que el concepto de RA se remonta a la década de 1960, el primer sistema formal de RA no se desarrolló hasta los años 90 por la compañía Boeing. A partir de entonces, se han ido llevando a cabo distintas conferencias sobre el tema incluyendo simposios internacionales sobre la RA o sobre los mundos virtuales, y los investigadores se han visto atraídos por las posibilidades de esta tecnología (Feiner, Macintyre, & Seligmann, 1993).

La definición más popular sobre RA es la dada por Milgram y Kishino (1994) quienes indican que "entre un entorno real y un entorno virtual puro está la llamada realidad mixta y esta se subdivide en 2, la realidad aumentada (más cercana a la realidad) y la virtualidad aumentada (más próxima a la virtualidad pura)" (Hsiao & Rashvand, 2011).

Por lo tanto la RA es un sistema interactivo que tiene como entrada la información del mundo real y superpone a la realidad nueva información digital en tiempo real, esta información virtual puede ser imágenes, objetos 3D, textos, videos etc. (Arribas, Gutiérrez, Gil, & Santos, 2014). Durante este proceso, la percepción y el conocimiento que el usuario tiene sobre el mundo real se ve enriquecido (MIT, 2009).

En educación, la capacidad para simular situaciones y experiencias que no sean posibles en el mundo real permite el aprendizaje de una forma más intuitiva e interactiva, por lo que la realidad aumentada puede ser una herramienta alternativa en la enseñanza y en la superación de estas dificultades, ya que permite que los estudiantes experimenten pensamientos, emociones y conductas similares a las que viven en una situación en la vida real. Pero para poder aprovechar el gran potencial de la realidad aumentada en los procesos de aprendizaje tiene que estar cuidadosamente en sintonía con los modelos pedagógicos y los estilos de aprendizaje individuales de los estudiantes que participen en una experiencia de aprendizaje específica (Fabregat Gesa, 2012).

Implementación caso de estudio

La investigación planteada corresponde a un pre experimento aplicado en dos grupos diferentes cuyo diseño general G1 X O1 y G2 - O2 tomó como grupo

experimental (G1) los 17 estudiantes de la asignatura de Introducción a la Programación de primer semestre de Ingeniería de Sistemas de la institución universitaria CESMAG (U1) y a los 16 estudiantes de Lógica Computacional del programa de Ingeniería Electromecánica de la Universidad Antonio Nariño sede Pasto (U2), orientada también en primer semestre. Ambos grupos correspondientes al periodo académico II-2014, y a los que se aplicó el tratamiento (X) que consistió en un ambiente demostrativo construido con técnicas de realidad aumentada y soportada con aprendizaje colaborativo como estrategia didáctica para el aprendizaje de estructuras iterativas y, finalmente, a los que se aplicó una prueba posterior (O1) al tratamiento.

Por otro lado, se tomó como grupo control (G2) a los 31 estudiantes de la asignatura de Introducción a la Programación de primer semestre de Ingeniería de Sistemas de la Institución Universitaria CESMAG (U1) y a los 27 estudiantes de Lógica Computacional del programa de Ingeniería Electromecánica de la Universidad Antonio Nariño sede Pasto (U2), ambos grupos correspondientes al periodo académico I-2014 y a los que no se aplicó tratamiento experimental.

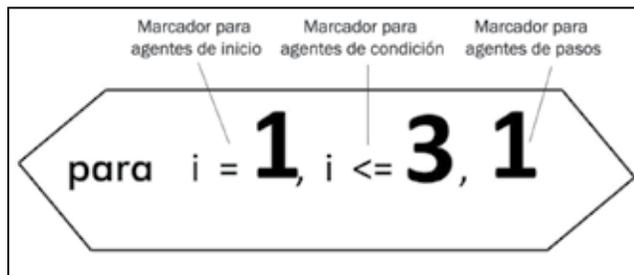
Debido a que las funciones que debe realizar el Diseñador Instruccional corresponden a crear ambientes interesantes de aprendizaje y actividades para encadenar la nueva información con el conocimiento previo, brindando oportunidades para el trabajo colaborativo y ofreciendo a los estudiantes una variedad de tareas de aprendizaje auténticas (C. Collazos et al., 2012), en cada grupo de estudiantes se crearon los siguientes roles para abordar los conceptos de la "estructura iterativa para": un primer grupo de estudiantes era considerado agentes de inicio, el segundo grupo como agentes de fin y un tercer grupo como agentes de pasos.

La estrategia colaborativa llevada a efecto para esta investigación consistió en dos etapas: la primera de configuración inicial llevó a efecto las siguientes actividades: definición de objetivos individuales y particulares, diseño de ejercicios y tareas, estimación del tiempo para realización de ejercicios y tareas, planeación de recursos y materiales necesarios para las actividades propuestas, conformación de los equipos de trabajo y planeación de la distribución física de los estudiantes en cada aula de clase. En la segunda

etapa, durante todo el desarrollo de la investigación, se garantizó la aplicación de los principios básicos del proceso colaborativo como: la interdependencia positiva, la responsabilidad individual, el desarrollo y practica de habilidades cognitivas e interpersonales, la interacción simultánea y los procesos de evaluación y reflexión personal y colectivo.

Así, una vez definidos los roles se generaron, para cada uno de ellos, impresos en papel cartulina los marcadores que se presentan en las figura 1.

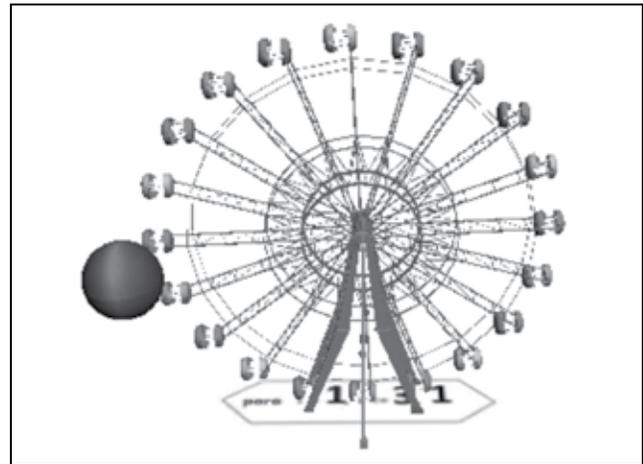
Figura 1. Marcadores por rol de agente



Tal como lo muestra la figura 1, en el proceso experimental se imprimieron marcadores para el rol de agentes de inicio con valores como: $i = 1$, $i = 3$, $i = 5$, de igual manera para el rol de agentes de condición se imprimieron marcadores como: $i \leq 3$, $i \leq 5$, $i > 2$, y en el caso del rol para agentes de pasos se contó con las impresiones de valores como: 1 , 3 , -1 entre otros.

Con los valores anteriores se solicitó a los estudiantes que formaran la estructura del ciclo, para que (previa explicación y documentación de su sintaxis) con sus tres parámetros y mediante la instalación de un APK en dispositivos móviles como celulares y tablet con sistema operativo Android, se realizara la visualización de la incidencia de los parámetros en un ambiente tridimensional, en el que se simulaba la configuración de dicho ciclo. En la figura 2 se observa la acción de la simulación del ciclo configurado para realizar 3 vueltas de la rueda de chicao, debido a que los parámetros de la estructura iterativa consisten en que inicien en el valor 1 hasta 3 con pasos de uno en uno.

Figura 2. Simulación caso éxito



De igual forma, en la figura 3 se presenta la incidencia de organizar erróneamente los parámetros en la estructura iterativa, lo que conlleva a que no se pueda realizar iteraciones en dicho ciclo.

Figura 3. Simulación caso sin iteraciones



Análisis y discusión de resultados

Una vez realizado el tratamiento investigativo en el grupo experimental (G1 conformado por dos grupos de las instituciones U1 y U2), cuyo objetivo principal era ilustrar al estudiante en la operatividad funcional que tiene la implementación de una estructura iterativa mediante técnicas de realidad aumentada y soportada por aprendizaje colaborativo como estrategia didáctica, se procedió a la realización tradicional de la clase de fundamentos de programación, en la que mediante una serie de ejercicios adecuados se estudia la incidencia de las estructuras iterativas en el ámbito de la programación.

En la tabla 2 se observan los resultados académicos de cada grupo después del tratamiento experimental y, al mismo tiempo, se los compara con los registros de notas obtenidos en un curso anterior con el mismo docente pero con metodología tradicional impartida en clase.

Tabla 2. Resultado de post prueba grupo experimental y control.

Institución	Periodo académico	Grupo	Promedio de notas
U1	I-2014	Control	4,23
	II-2014	Experimental	4,53
U2	I-2014	Control	4,21
	II-2014	Experimental	4,67

Como se aprecia en la tabla 2 los resultados obtenidos por el grupo experimental demuestran la incidencia que generó el tratamiento investigativo, ya que en cierta manera y a través de la simplicidad de un ejemplo elemental, combinado con dos herramientas adecuadas (Realidad aumentada y Aprendizaje Colaborativo) como estrategia didáctica de aprendizaje, se logró que el estudiante asocie de una manera vivencial la aplicación de las estructuras iterativas en el proceso de estudio, en el caso de la implementación de ciclos en el pensamiento y práctica computacional.

Conclusiones y trabajo futuro

La incorporación de procesos tecnológicos llamativos, como lo es la realidad aumentada combinada con metodologías de aprendizaje basadas en colaboración, concluyen en una novedosa estrategia didáctica como apoyo frente a la compleja tarea de guiar al estudiante de desarrollo de software en la adquisición de competencias sólidas, desde los procesos tempranos relacionados con el estudio de los fundamentos de programación.

Como trabajo futuro se contempla la inclusión del estudio de estructuras condicionales y demás conceptos iterativos, que finalmente puedan concluir en la construcción de una metodología que, utilizando aprendizaje colaborativo y combinado con procesos inmersivos 3D, permitan que el estudiante de fundamentos de programación incorpore tales aprendizajes de una manera más vivencial, aprovechando las bondades tecnológicas y la eficiencia de la colaboración como proceso inherente al comportamiento de los seres humanos.

Referencias

Affleck, G. and Smith, T. (1999). Identifying a need for web - based course support.

Arribas, C., Gutiérrez, M., Gil, C., & Santos, C. (2014). Recursos digitales autónomos mediante realidad aumentada, 17, 241–274.

Association for Computing Machinery, I. C. S. (2009). Computer Science Curriculum 2008: An Interim Revision of CS 2001. 8 de Marzo de 2009.

Ben-Ari, M. (2001). Journal of Computers in Mathematics & Science Teaching, 20, 24–73.

Bett, G., Fernández, J., Frittelli, V., Serrano, D., Steffolani, F., Strub, A., ... Teicher, R. (2013). Desarrollo de Juegos como Estrategia Didáctica en la Enseñanza de la Programación. Universidad Tecnológica Nacional, Facultad Regional Córdoba. Retrieved May 22, 2014, from <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/4/120-429-1-DR.pdf>

Buriticá, T., & Ivan, O. (2012). Consideraciones sobre la evolución del pensamiento a partir de los paradigmas de programación de computadores. Redalyc.

Collazos, C. A., & Mendoza, J. (2006). How to take advantage of " cooperative learning " in the classroom. Educación Y Educadores, 9(Universidad de la Sabana), 61–76.

Collazos, C., Guerrero, L., & Vergara, A. (2012). Aprendizaje Colaborativo: un cambio en el rol del profesor.

De la Torre Zermeño, F. (2005). 12 lecciones de pedagogía, educación y didáctica. (Alfaomega, Ed.) (Alfaomega, p. 215). México: Alfaomega.

Díaz Barriga, F., & Hernández Rojas, G. (1999). Estrategias Docentes para un Aprendizaje Significativo: Una interpretación constructivista. (McGraw Hill & S. A. Interamericana, Eds.) (p. 215). México: McGrawHill Interamericana, S.A.

- Fabregat Gesa, R. (2012). Combinando la realidad aumentada con las plataformas de e-learning adaptativas. *Revista Venezolana de Información, Tecnología Y Conocimiento*, 9, 69–78.
- Feiner, S., Macintyre, B., & Seligmann, D. (1993). Knowledge-based augmented reality. *ACM*, 36(7), 53–62.
- Fonseca, M. C., & Aguaded Gómez, J. I. (2007). Enseñar en la universidad. Experiencias y propuestas de docencia universitaria (Netbiblo., p. 268). España.
- González De Rivera Fuente, M., & Paredes Velasco, M. (2008). Aprendizaje con programación Colaborativa, 1–33.
- Hernández, G., Jiménez, R., & Martínez, Á. (2012). Creencias docentes sobre la importancia de la didáctica en la orientación de la enseñanza del primer curso de programación de computadoras. Editorial Académica Española. Retrieved May 22, 2014, from file:///C:/Users/DaSar/Downloads/Creencias docentes sobre la importancia de la did% C3%A1ctica en la orientaci% C3%B3n de la ense% C3%B1anza del primer curso de programaci% C3%B3n de computadoras.pdf
- Hsiao, K., & Rashvand, H. F. (2011). Body Language and Augmented Reality Learning Environment. Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering, 246–250.
- Jonhson, D. W., Jonhson, R., & Holubec, E. (1993). *Circles of learning* (4th ed.).
- Millis, B. (1996). Materials presented at The University of Tennessee at Chattanooga Instructional Excellence Retreat. Chattanooga Instructional Excellence Retreat, (University of Tennessee).
- MIT. (2009). *Faster Maintenance with Augmented Reality*. Technology Review, MIT.
- Monterrey, I. T. y de E. S. de. (2004). *Aprendizaje Colaborativo, estrategias y técnicas didácticas en el rediseño*. Dirección de Investigación Y Desarrollo Educativo.
- Monterrey, I. T. y de E. S. de. (2008). *Aprendizaje Colaborativo, técnicas didácticas*. Programa de Desarrollo de Habilidades Docentes, (Dirección de investigación e innovación educativa), 16.
- Mosser, R. (1997). In anual In Annual Joint Conference Integrating Technology into Computer Science Education ACM Press New York, NY, USA, 114–116.
- Murillo Rivera, M. (2006). Exploring the teaching-learning process. *actualidades investigativas en educacion explorando*, 6, 0–28.
- Oviedo, M. (2002). *La enseñanza de la programación*. UPIICSA, México. Retrieved May 21, 2014, from <http://bibliotecadigital.conevyt.org.mx/colecciones/documentos/somece2002/Grupo4/Oviedo.pdf>
- Szpiniak, A. F., & Rojo, G. A. (2007). *Enseñanza de la programación*.
- Truong, N., Bancroft, P., & Roe, P. (2003). *A Web Based Environment for Learning to Program*, 16.