



ESTRATEGIA COLABORATIVA EN ENTORNOS TRIDIMENSIONALES COMO ESTRATEGIA DIDÁCTICA DE APRENDIZAJE DE ESTRUCTURAS ITERATIVAS EN PROGRAMACIÓN COMPUTACIONAL¹

Recibido: abril 09 de 2015/ **Revisado:** noviembre 18 de 2015/ **Aceptado:** noviembre 5 de 2015
Por: Javier Alejandro Jiménez Toledo², Cesar Alberto Collazos Ordoñez³,
Julio Ariel Hurtado Alegría⁴ y Wilson Libardo Pantoja Yépez⁵

Para citar este artículo/To reference this article/Para citar este artigo

Jiménez, J., Collazos, C., Hurtado, J. & Pantoja, W. (Julio- diciembre, 2015). Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional. *Investigium IRE: Ciencias Sociales y Humanas*, VI (2), 80-92. Doi: <http://dx.doi.org/10.15658/CESMAG15.05060207>

RESUMEN

Este artículo presenta una nueva propuesta didáctica que surge a partir de un proceso investigativo cuyo propósito planteado pretende facilitar el estudio de las estructuras iterativas en el área de los fundamentos de programación, ofreciendo al estudiante conceptos complejos en un contexto apropiado a su pre saber. La investigación se desarrolló bajo un enfoque cuantitativo con el método empírico analítico y cuyo diseño se realizó de manera cuasi experimental en dos universidades con dos grupos de control y dos experimentales de estudiantes de pregrado de primer semestre de Ingeniería quienes fueron evaluados mediante aplicación de post pruebas después de recibir o no tratamiento experimental el cual consistió en la aplicación de escenarios apoyados con procesos de realidad aumentada con objetos 3D en un contexto colaborativo. Los datos obtenidos fueron analizados con la distribución de probabilidad t de Student con la que se comprobó que la diferencia de notas entre grupo experimental y grupo de control para cada universidad participante es estadísticamente significativa, lo que concluyó el éxito del tratamiento experimental.

Palabras Clave: Aprendizaje colaborativo, estructuras iterativas, fundamentos de programación, realidad aumentada.

¹ Artículo derivado del proyecto de investigación "Propuesta metodológica basada en modelado gráfico para la enseñanza de los fundamentos de programación mediante un entorno colaborativo inmersivo 3D", inscrito en el grupo de investigación TECNOFILIA de la Institución Universitaria CESMAG y en el grupo de investigación IDIS de la Universidad del Cauca como tesis del programa de Maestría en Computación.

² Candidato a Magíster en Computación Universidad del Cauca, Especialista en Docencia Universitaria, Ingeniero de Sistemas y Computación, Profesor tiempo completo, Programa de Ingeniería de Sistemas, Institución Universitaria CESMAG, Pasto (Colombia). Correo electrónico: jajimenez@iucsmag.edu.co.

³ PostDoctorado Ciencias de la Computación, Doctor en Ciencias mención Computación, Ingeniero de Sistemas, profesor tiempo completo, Departamento de Sistemas, Universidad del Cauca, Popayán (Colombia). Correo electrónico: ccollazos@unicauca.edu.co.

⁴ Doctor en Ciencias mención Computación, Especialista en redes y servicios telemáticos, Especialista en procesos para desarrollo de software, Ingeniero en electrónica y telecomunicaciones, profesor tiempo completo, Departamento de Sistemas, Universidad del Cauca, Popayán (Colombia). Correo electrónico: ahurtado@unicauca.edu.co.

⁵ Magíster en Computación, Especialista en redes y servicios telemáticos, Ingeniero de Sistemas, profesor tiempo completo, Departamento de Sistemas, Universidad del Cauca, Popayán (Colombia). Correo electrónico: wpantoja@unicauca.edu.co.



COLLABORATIVE STRATEGIES IN THREE-DIMENSIONAL ENVIRONMENTS, AS STRATEGY DIDACTIC OF LEARNING OF ITERATIVE STRUCTURES IN COMPUTER PROGRAMMING

ABSTRACT

The article presents a new didactic proposal that emerges from an investigative process whose posed purpose is intended to facilitate the study of structures iterative in the area of the fundamentals of programming, offering to the students, complex concepts in a context appropriate to their previous knowledge. The research was developed under a quantitative approach, analytical empirical method and design quasi-experimental with two control groups and two experimental of first semester of Engineering undergraduate students from two universities, those who were evaluated through application of post tests after receiving or not experimental treatment, which consisted in the application of scenarios supported by processes of augmented reality with 3D objects in a collaborative context. The data obtained were analyzed with the probability distribution t Student that proved that notes difference between experimental group and control group for each participating University, is statistically significant, which concluded the success of the experimental treatment.

Key Words: learning collaborative, iterative structures, fundamentals of programming, augmented reality.

ESTRATÉGIAS DE COLABORAÇÃO EM AMBIENTES TRIDIMENSIONAIS COMO ESTRATÉGIA DIDÁTICA DE APRENDIZAGEM DE ESTRUTURAS ITERATIVAS EM PROGRAMAÇÃO DE COMPUTADORES

RESUMO

O artigo apresenta uma nova proposta didática que emerge de um processo investigativo cujo objetivo planejado é destinado a facilitar o estudo das estruturas interativas na área dos fundamentos de programação, ao oferecer ao aluno conceitos complexos num contexto adequado à sua pré-saber. A pesquisa foi desenvolvida sob uma abordagem quantitativa, método empírico analítico e design quase-experimental com dois grupos de controle e dois experimentais de alunos de graduação de primeiro semestre de Engenharia de duas universidades que foram avaliados através da aplicação de post testes depois de receber ou não tratamento experimental, o qual consistiu na aplicação de cenários suportados com processos de realidade aumentada com objetos 3D em um contexto colaborativo. Os dados obtidos foram analisados com a distribuição de probabilidade de Student com a que se verificou que a diferença de avaliações entre o grupo experimental e o grupo de controle para cada universidade participante, é estatisticamente significativa, o que concluiu o sucesso do tratamento experimental.

Palavras-Chave: aprendizagem colaborativa, estruturas iterativas, fundamentos de programação, realidade aumentada.



INTRODUCCIÓN

La enseñanza de los fundamentos de programación es un campo de estudio que ha tomado gran importancia debido al auge del software en la actualidad en todos los procesos del accionar humano, hasta tal punto de estar presente en dispositivos con *wearablethechnology* que es una sofisticada tecnología que se puede llevar puesta como otra prenda de vestir (Vidal, 2014). Pese al estado tecnológico actual existen aún problemas relacionados con la fundamentación de los futuros constructores de software cuyo origen se inicia 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, 2013); tales inconvenientes se presentan por múltiples situaciones que van desde la carencia de habilidades para programar, el desconocimiento de la materia y hasta la insuficiencia de disciplina en programación (Oviedo, 2002).

Muchos autores han propuesto diversas estrategias para el aprendizaje de los fundamentos de programación, es así como en los primeros cursos algunos docentes inician esta enseñanza, 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).

Una de las temáticas más compleja en la enseñanza de los fundamentos de programación es la relacionada con las estructuras iterativas debido a su complejidad estructural y más aún por la falta de asociación vivencial desde el punto de vista de un estudiante. Aprender a programar es reconocido por ser difícil, una de las mayores dificultades radica en que los lenguajes son artificiales en todas las plataformas de programación (Mosser, 1997). Truong, Bancroft y Roe (2007) 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 a 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.

En el estudio realizado por Affleck y Smith (1999) 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. 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.

Para Manuel Gonzales y Maximiliano Paredes (2008) 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. 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).

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. A continuación se presenta un listado de las diversas metodologías vigentes que sirven como herramientas de apoyo en la consolidación de las competencias necesarias que un desarrollador de software debe tener en su formación:

Trabajo Colaborativo como Estrategia para Mejorar el Proceso de Enseñanza-Aprendizaje – Aplicado a la Enseñanza Inicial de Programación en el Ambiente Universitario. Presenta la incorporación de herramientas que contribuyan al aprendizaje colaborativo en la enseñanza de programación (Hernández et al., 2012).

EDCIA: El entorno ECDIA está basado en una herramienta que integra el trabajo con metodologías de diseño de algoritmo y lenguajes de programación, su principal aporte es adicionar la posibilidad de trabajo colaborativo en la edición de algoritmos (Fracchia, Baeza & Martins, 2012).

VPL. Es un gestor de prácticas de programación sobre Moodle, posibilitando la entrega, edición y ejecución de las prácticas de programación, además de una evaluación continua y automática de estas (Hernández et al. 2013).

Entorno de integración de PBL y CSCL para la enseñanza de algoritmos y programación en ingeniería. Denominado SÁBATO el cual es utilizado en la enseñanza de la asignatura algoritmos y programación en ingeniería. (Jiménez, Panovi & Álvarez, 2008).

SCRATCH. Es un lenguaje de programación que facilita crear sus propias historias interactivas, animaciones, juegos, música y arte. A medida que los jóvenes crean y comparten proyectos realizados en Scratch, aprenden importantes ideas matemáticas y computacionales (MIT, 2008). Scratch se caracteriza por: es programación orientada a objetos, existencia de un banco de recursos en el propio programa, visualización instantánea de lo que hacemos en el programa; interacción con los objetos y personajes, posibilidad de difundir nuestro proyecto en la web (Colmenar, 2011).

Cupi2. Es un proyecto de la Universidad de los Andes que tiene como objetivo la búsqueda de nuevas maneras para enfrentar el problema de enseñar a programar (Villalobos, 2007). Cupi2 como proyecto de investigación se encuentra soportado por cuatro componentes que permiten la construcción de una solución balanceada frente problema de aprender a programar (De Graff & Kolmos, 2007).

Una Herramienta y Técnica para la Enseñanza de la Programación. Esta línea de investigación enfatiza los diversos enfoques de la educación que se encuentran en boga, el estilo de aprendizaje de los alumnos y los conceptos técnicos de programación, sustentándolo, en una técnica para el diseño de algoritmos y una herramienta automatizada en el desarrollo de programas (Calderón, 2008).

GREENFOOT. Es un Mundo Visual Interactivo, enseña la orientación a objetos con Java. Crea



“actores” que viven en “mundos” para construir juegos, simulaciones y otros programas gráficos (Kolling, 2010).

JeCo. Los estudiantes pueden adquirir vocabulario relacionado con los conceptos de programación de Jeliot y así se crea un contexto para las discusiones, en el caso de JeCo, este vocabulario se convierte en un conjunto inter-subjetivo de conceptos compartidos (Myller and Nuutinen, 2006).

ProLearn. Puede promover la programación dinámica concreta de un programa y visualmente apoyar el trabajo práctico en el propio ritmo de aprendizaje del estudiante (Mendes, 2006). Además, es una herramienta que ayuda a los programadores noveles a resolver problemas usando comunicación basada en texto (González & Fuentes 2008).

PL-Detective. Permite aprender nociones de programación de manera colaborativa (González & Fuentes 2008), fue diseñado para la construcción asignaciones y demostraciones en el contexto de un curso titulado “Principios de lenguajes de programación”. El PL-Detective es una implementación flexible y extensible del lenguaje llamado “Mystery” (Diwan, Waite, & Jackson et al., 2005).

COLLEGE: En este proyecto se modela una propuesta de sistema colaborativo síncrono para el aprendizaje de la programación que pueda ser utilizado tanto en los laboratorios de prácticas como desde otros espacios (Redondo, 2004).

OOP ANIM. Ha sido desarrollado para ayudar a nuestros estudiantes a aprender los conceptos básicos de la programación orientada a objetos y al desarrollar sus capacidades de programación que utiliza este paradigma. (Esteves & Mendes, 2004).

DPE. Es el Entorno de Programación Distribuida (DPE) donde los programadores y gerentes de proyectos (o profesores y estudiantes) pueden ser ubicados en cualquier lugar y efectivamente

realizar procesos colaborativos una vez que se conectan entre sí a través de Internet (Jo & Arnold, 2003).

ELP. Es un entorno que permite a los estudiantes programar en sus primeros cursos sin la necesidad de familiarizarse con sintaxis de lenguajes de programación, lo más importante es que elimina todas las dificultades asociadas con la instalación y ejecución de un compilador de Java (Truong et al., 2007)

BlueJ. Es un Entorno de desarrollo Integrado (IDE) específicamente desarrollado para enseñar y aprender Programación Orientada a Objetos y presenta un enfoque pedagógico desarrollado para ser utilizado con un sistema como el de Java (Kölling, Quig, Patterson & Rosenberg, 2003).

ALICE. Es un programa para construir mundos virtuales con objetos 3D, al ver en forma inmediata cómo corren los programas de animación, los estudiantes pueden entender con mayor facilidad la relación entre el código y el comportamiento de un objeto (University C.M., 2003).

HABIPRO. Establece el cómo el aprendizaje cooperativo apoyado en computador (CSLC) se puede implementar para desarrollar nuevas habilidades en estudiantes universitarios que condujeron al desarrollo de un ambiente adaptativo para el manejo de buenos hábitos de programación (Vizcaíno, Contreras, Favela. & Prieto, 2002).

ALGOARENA. Es un software educativo para el aprendizaje de programación. AlgoArena es un sistema educativo de colaboración a través del cual los estudiantes pueden aprender a programar, participando conjuntamente en la formación de una comunidad de práctica y su identidad en ella (Suzuki & Kato, 1997).

SIGACLE. Simulador gráfico asistido por computador para la enseñanza de programación de lenguajes estructurados, el objetivo pedagógico es presentar al estudiante vistas diferentes de lo que sucede en la máquina al ejecutar un programa. Al principio, el estudiante no es consciente



de la importancia de utilizar metodologías para el desarrollo de programas [Humphrey 1997], por lo general, él se dedica a escribir código de forma artesanal, dejando el problema de corrección al compilador (Aho & Ullman, 1973).

Por lo anterior, la presente investigación tomó como base las teorías relacionadas con aprendizaje colaborativo, estrategias didácticas y realidad aumentada como fundamentos interrelacionados para lograr el propósito. Es así como 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 (Collazos, Guerrero, & Vergara, 2012).

En el mismo sentido, la implementación de actividades colaborativas con el fin de mejorar el proceso en enseñanza aprendizaje en el aula incluye básicamente dos acciones importantes en dicho proceso: una etapa de configuración de las condiciones iniciales y una segunda etapa que garantice la aplicación de los principios básicos de una práctica colaborativa (Collazos & Mendoza, 2006). En tal sentido, en el rol de profesor como Diseñador Instruccional, el profesor 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 (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, 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 & Hernández, 1999).

A pesar de que el concepto de Realidad Aumentada (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. (Feiner, Macintyre, & Seligmann, 1993). La definición más popular sobre RA es la dada por Milgram y Kishino 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, p. 246).

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 pueden 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).

METODOLOGÍA

La investigación planteada se enmarca en el enfoque cuantitativo y corresponde a un cuasi experimento aplicado en cuatro grupos diferentes pertenecientes a dos instituciones de educación superior del municipio de Pasto (Nariño), para lo cual se planteó un diseño experimental para cada institución según el siguiente modelo:

Tabla 1. Diseño experimental

Institución	Diseño Experimental
Institución Universitaria CESMAG (I.U.CESMAG)	$G_1 \times O_1$ $G_2 - O_2$
Universidad Antonio Nariño (UAN)	$G_3 \times O_3$ $G_4 - O_4$

El primer grupo experimental (G_1) fue conformado por 17 estudiantes de la asignatura de Introducción a la programación de primer semestre de Ingeniería de Sistemas correspondientes al periodo académico II-2014 a quienes se les aplicó el tratamiento (X) que consistió 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 quienes se le aplicó una prueba posterior (O_1) al tratamiento. El grupo de control (G_2) estuvo conformado por 30 estudiantes del periodo académico I-2014 del mismo semestre y asignatura del grupo experimental a quienes no se les aplicó tratamiento experimental y las notas obtenidas del tema de estudio fueron consideradas como O_2 .

El segundo grupo experimental (G_3) estuvo constituido por 17 estudiantes de la asignatura de Lógica Computacional de primer semestre de Ingeniería Electromecánica correspondientes al periodo académico II-2014 a quienes también se les aplicó el mismo tratamiento (X) con su correspondiente evaluación (O_3) y su grupo de control (G_4) fueron los 27 estudiantes del periodo académico I-2014 del mismo semestre y asignatura del grupo experimental a quienes tampoco se les aplicó tratamiento

experimental, considerando también las notas obtenidas del tema en estudio (O_4).

Un ciclo o también llamado estructura iterativa o de repetición en programación de computadores permite ejecutar una o varias instrucciones, un número determinado de veces o, indefinidamente, mientras se cumpla una condición (López, 2009), los ciclos más importantes y estudiados son: ciclo para, ciclo mientras y ciclo hacer mientras. Para comprender mejor el concepto de ciclo imagine una máquina de tren que parte de la estación "inicio" y viaja hasta la estación "final" (Fidalgo, 2012); cuando el maquinista llega a la estación "final" evalúa mediante una condición si vuelve o no a la estación "inicio" y así sucesivamente hasta que en dicha condición ya no se determine regresar a estación "inicio". Cada vuelta que da el maquinista es una ejecución o una iteración, en la cual se ejecutan todas las instrucciones que hay en el trayecto comprendido entre la estación "inicio" y la estación "final". La figura 1 muestra este concepto.

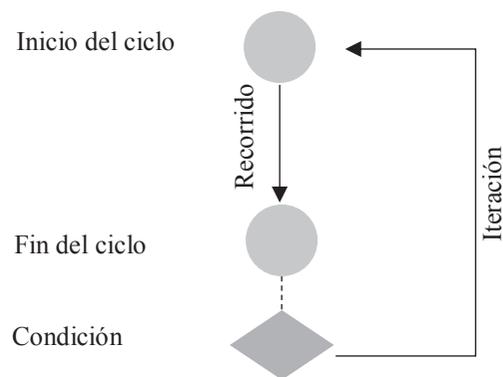


Figura 1. Ejemplo de ciclo

La ruta metodológica llevada a efecto para la investigación consistió en dos etapas:

Configuración inicial: en esta etapa se llevaron a efecto las siguientes actividades como 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.

Desarrollo de la investigación: es esta etapa 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.

El modelado colaborativo utilizado empleó dos roles: docente y estudiante. El docente asumió características de mediador cognitivo, instructor y diseñador instruccional para la realización de una configuración inicial, la cual tuvo dos momentos: la primera fue a nivel de la orientación al grupo de estudiantes respecto de la fundamentación conceptual de la temática bajo un modelado gráfico y en las segunda se realizaron casos de estudio utilizando el método JigSaw, donde el estudiante participó en tres roles y estuvo orientado por los principios colaborativos que le permitieron interactuar con los demás compañeros en las actividades planteadas.

La estrategia didáctica para la aplicación del tratamiento experimental de la investigación (X) comenzó dividiendo en cada institución participante al grupo en tres subgrupos de estudiantes (los participantes de G_1 se dividieron en tres sub-

grupos lo mismo con G_3) para los cuales se crearon los siguientes roles para abordar los conceptos de "estructura iterativa para" (ver figura 2): un primer subgrupo de estudiantes era considerado agentes iniciadores, el segundo como agentes de fin y un tercero como agentes de pasos. Así, una vez definidos los roles se generó para cada uno de ellos impresos en papel los marcadores que se debían incorporarse en la plantilla base (ver figura 2).

En la figura 2, en el proceso experimental se imprimió 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 $i <= 3, i <= 5, i >= 2$ y en el caso del rol para agentes de pasos se contó con las impresiones de valores como 1, 3, -1 entre otros. Una vez conformado el grupo inicial de estudiantes (que constaba de 3 compañeros) y apoyados del método JigSaw se le asignó a cada uno un rol, luego cada alumno se reunió con pares de otro grupo con el propósito de debatir y elegir la configuración adecuada de acuerdo al caso de estudio planteado, este proceso se repitió hasta formar tres grupos totales conformados por cada rol (un solo grupo de agentes iniciadores, otro de agentes de fin y un último de agentes de paso), asegurando un diálogo entre pares; posteriormente cada estudiante regresó a su grupo inicial para dar solución al ejercicio planteado y cuyo diagrama de actividades se aprecia en la figura 3.

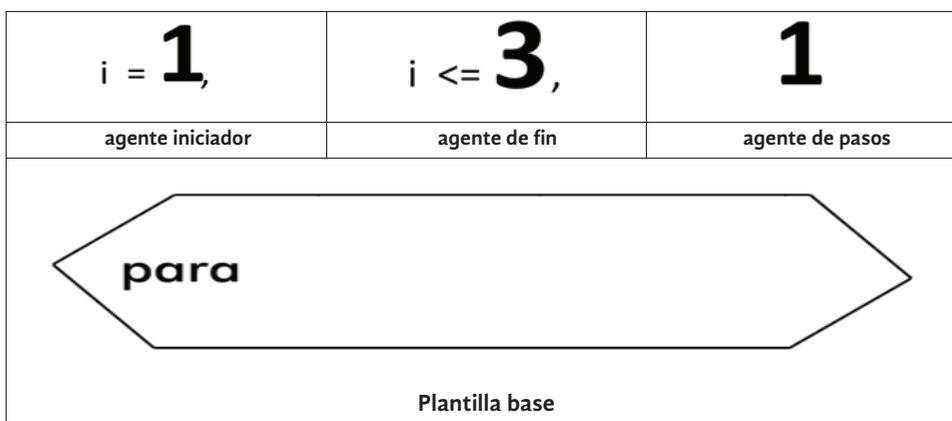


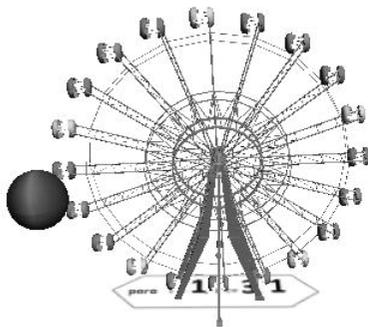
Figura 2. Plantillas y roles para estudiantes



Figura 3. Diagrama de actividades para caso de estudio

Una vez los estudiantes eligieron las configuraciones adecuadas de acuerdo al rol asignado y con los elementos de cada agente impresos en papel, se solicitó a cada grupo de estudiantes que formen la estructura del “ciclo para” con una previa explicación y documentación de su sintaxis y mediante la instalación de un APK en los dispositivos móviles de los estudiantes tales como celulares y Tablet se realizó la visualiza-

ción de la incidencia de los parámetros en un ambiente tridimensional en el que se simulaba la configuración de dicho ciclo. En la figura 4 se observa la acción de la simulación del ciclo configurado en la que en el caso éxito, el resultado consiste en la visualización de 3 vueltas de la rueda de chicao debido a la configuración apropiada y en el caso error se presenta debido a la incorrecta organización de los parámetros.



Simulación caso éxito

Ciclo no realiza iteraciones

Simulación caso error

Figura 4. Simulaciones

RESULTADOS Y DISCUSIÓN

Una vez aplicado el tratamiento investigativo en cada grupo experimental (G_1 y G_3) cuya estrategia didáctica se basó en la incorporación de técnicas de realidad aumentada soportada por aprendizaje colaborativo como fase de sensibilización al concepto de estructura iterativa utilizando el “ciclo para”, se

procedió a la realización tradicional de la clase en la que mediante una serie de ejercicios adecuados se estudia la incidencia de las estructuras iterativas en el ámbito de la programación. Posterior a la fundamentación se realizó para cada grupo (G_1 , G_2 , G_3 y G_4) dos actividades evaluativas que consistieron en la aplicación de dos talleres grupales con la participación de dos estudiantes (40%) y dos seguimientos individuales (60%) con el propósito de establecer cuantitativamente el proceso de apropiación de dicha temática y cuyos resultados se aprecian en la tabla 2.



Tabla 2. Promedio de notas grupos experimentales y control

Grupo	Periodo	Promedio
G1	II-2014	4,5
G2	I-2014	3,7
G3	II-2014	4,7
G4	I-2014	3,6

Con los resultados obtenidos tanto para los grupos experimentales como de control se realizó un análisis estadístico mediante la distribución de probabilidad *t de Student* con un nivel de confianza del 95% cuyos valores se encuentran en la tabla 3.

La tabla 3 evidencia el resultado de la aplicación del tratamiento experimental en cada una de la instituciones de educación superior, los resultados estadísticos obtenidos en ambos grupos experimentales (G_1 y G_3) adquieren un valor estadístico *t* (2,3412 y 3,4898) mayor tanto al valor crítico de *t* de una cola (1,6794 y 1,6820) como al valor crítico para dos colas (2,0141 y 2,0181) y el valor de *P* es menor a 0,05 (5%) en los cuatro casos (*P* para una y dos colas) lo cual concluye que la diferencia de notas entre grupo experimental y grupo de control

para cada institución es estadísticamente significativa para $t=5\%$.

Por lo tanto, el análisis estadístico anterior demuestra la incidencia que tiene el tratamiento presentado en esta investigación en los grupos experimentales frente a los grupos de control, estableciendo que al incorporar estrategias didácticas adecuadas en la complejidad de una temática y asociándola con los pre saberes de una manera apropiada se obtienen resultados académicos que benefician de una manera directa a los estudiantes. Cabe mencionar que los resultados obtenidos permitieron mejorar el nivel de apropiación de las temáticas posteriores referentes al estudio de las demás estructuras cíclicas debido a la buena percepción del manejo del primer ciclo, en este caso el "ciclo para".

El proceso colaborativo llevado a cabo en la investigación permitió que los estudiantes se inter relacionara de forma activa, logrando así una dinámica interactiva que mejoró las relaciones interpersonales del grupo, que por cierto fue sencilla de realizar dadas las características de los grupos experimentales por tratarse de estudiantes de primer semestre de pregrado en los cuales aún no

Tabla 3. T de Student por grupos

Conceptos	I.U. CESMAG		UAN	
	G1	G2	G3	G4
Media	4,5300	3,7067	4,6731	3,5674
Varianza	0,3954	1,8642	0,1635	1,5910
Observaciones	17,0000	30,0000	17,0000	27,0000
Varianza agrupada	1,3420		1,0472	
Diferencia hipotética de las medias	0,0000		0,0000	
Grados de libertad	45,0000		42,0000	
Estadístico <i>t</i>	2,3412		3,4898	
$P(T \leq t)$ una cola	0,0119		0,0006	
Valor crítico de <i>t</i> (una cola)	1,6794		1,6820	
$P(T \leq t)$ dos colas	0,0237		0,0011	
Valor crítico de <i>t</i> (dos colas)	2,0141		2,0181	



existe la diferenciación de grupos marcados que se presentan generalmente con el paso de tiempo. Esto permitió que los estudiantes se acoplaran fácilmente al rol establecido el cual fue dinámico durante los diversos casos de estudio planteados.

En el estudio se pudo evidenciar que al mezclar la rigurosidad de una teoría (en este caso el estudio de la "estructura iterativa para") con procesos que se encuentren dentro de los esquemas de acción de los estudiantes (escenarios tridimensionales soportados con realidad aumentada) y mediado con una didáctica de participación activa (estrategia colaborativa utilizando el método *JigSaw*), es más sencillo obtener por tiempos más prolongados la atención del estudiante, que involucrado en un proceso totalmente interactivo comienza a construir las bases de su lógica computacional cuya importancia es vital para los procesos que se requieren en posteriores semestres.

Finalmente y especialmente en el campo de la enseñanza para formación de futuros constructores de software, se hace necesario involucrar aspectos didácticos que contemplen acciones mediadas por tecnología que incluyan al estudiante de manera activa en la construcción de su lógica computacional, inculcándole la necesidad del autoaprendizaje en un ambiente colaborativo que mejore las actitudes y aptitudes del estudio y trabajo en equipo.

CONCLUSIONES

Las calificaciones obtenidas por los dos grupos experimentales fueron numéricamente superiores a las obtenidas por los grupos de control, lo cual demuestra la incidencia del tratamiento experimental llevado a cabo en el cual se combina adecuadamente aprendizaje colaborativo con técnicas de realidad aumentada.

La incorporación de procesos tecnológicos acordes al momento tecnológico vivenciado por los estudiantes, como lo es la realidad aumentada, combinada con metodologías de aprendizaje colaborativo, permitieron capturar fácilmente la atención del estudiante, experimentando así una

novedosa estrategia didáctica como apoyo frente a la compleja tarea de guiar al estudiante de desarrollo de software a 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. & Smith, T. (december, 1999). Identifying a need for web - based course support. *16th annual conference of the Australasian Society for Computers in Learning in Tertiary Education ASCILITE*. University of Paisley, Paisley, Brisbane (Australia).
- Aho, A. & Ullman, J. (1973). *The theory of parsing, translation and compiling Cliffs*. N. Jersey, E.E.U.U.: Prentice Hall.
- Arribas, C., Gutiérrez, M., Gil, C., & Santos, C. (febrero, 2014). Recursos digitales autónomos mediante realidad aumentada. *Revista iberoamericana de educación a distancia RIED*, 17, 241–274.
- Ben-Ari, M. (enero, 2001). Constructivism in Computer Science Education. *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. Recuperado de <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/4/120-429-1-DR.pdf>
- Calderón, R. (2008). *Una Herramienta y Técnica para la Enseñanza de la Programación*. (Tesis de maestría inédita) Universidad Politécnica del Valle de México. México.
- Colmenar, N. (agosto, 2011). Entornos para enseñar programación en secundaria. *Revista de Nuevas Tecnologías y Sociedad*, 1–13.
- Collazos, C. & Mendoza, J. (junio, 2006). How to take advantage of “cooperative learning” in the classroom. *Educación Y Educadores*, 9, 61–76.
- Collazos, C., Guerrero, L., & Vergara, A. (agosto, 2012). Aprendizaje Colaborativo: un cambio en el rol del profesor. *Tercer Congreso de Educación Superior en Computación*, Punta Arenas (Chile).
- De Graff, E. & Kolmos, A. (2007). *Management of Change. Implementation of Problem-Based and Projec-Based learning in Engineering*. Netherlands: Sense Publishers.
- De La Torre, F. (2005). *12 lecciones de pedagogía, educación y didáctica*. México: Alfaomega.
- Díaz, F. & Hernández, G. (1999). *Estrategias Docentes para un Aprendizaje Significativo: Una interpretación constructivista*. México: McGrawHill Interamericana.
- Diwan, A., Waite, W. & Jackson, M. (december, 2005). PL-Detective: A System for Teaching Programming. *Journal on Educational Resorces in Computing*, 1, 65-79.
- Esteves, M. & Mendes, A. (octubre, 2004). A simulation tool to help learning of object oriented programming basics. *34th Annual Frontiers in Education FIE*. 811–816.
- Feiner, S., Macintyre, B. & Seligmann, D. (agosto-diciembre, 1993). Knowledge-based augmented reality. *ACM*, 36(7), 53–62.
- Fidalgo, A. (2012). Unidad Didáctica No. 4. Bucles. *Universidad Politécnica de Madrid*. Recuperado de <https://innovacioneducativa.files.wordpress.com/2012/10/bucles.pdf>
- Fracchia, C., Baeza, N. & Martins, A. (octubre, 2012). ECDIA: Entorno Colaborativo para el Diseño e Implementación de Algoritmos. *Anales del XVIII congreso argentino de ciencias de la computación CACIC*, Bahía Blanca, Buenos Aires (Argentina).
- González, M. & Paredes, M. (febrero, 2008). Aprendizaje con programación Colaborativa. *Serie de Informes Técnicos DLSI1-URJC*, 1–33.
- Hernández, G., Jiménez, R. & Martínez, Á. (enero-junio, 2013). Creencias docentes sobre la importancia de la didáctica en la orientación de la enseñanza del primer curso de programación de computadoras. *Revista universitaria: docencia, investigación e innovación*, 2, 87-103.
- Hsiao, K. & Rashvand, H. (junio, 2011). Body Language and Augmented Reality Learning Environment. *Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering*, 246.
- Jiménez, J., Panovi, M. & Álvarez, A. (diciembre, 2008). Enseñanza de algoritmos y programación en ingeniería Integration environment of PBL and CSCL for teaching algorithms and programming in engineering. *Avances en Sistemas e Informática*, 5(3), 189-194.
- Jo, C. & Arnold, A. (mayo, 2003). A Portable and Collaborative Distributed Programming Environment The architecture of DPE. *International Conference on Software Engineering*, Las Vegas (EEUU).
- Jonhson, D., Jonhson, R. & Holubec, E. (1993). *Circles of learning 4th ed*. Poenix: Interaction Book Company
- Kolling, M. (November, 2010). The Greenfoot programming enviroment. *ACM*, 10,4-14.



- Kölling, M., Quig, B., Patterson, A. & Rosenberg, J. (enero, 2003). The Bluej system and its pedagogy. *Computer Science Education*, 13(4), 1–12.
- López, J. (2009). *Algoritmos y programación. Guía para docentes*. Recuperado de <http://www.eduteka.org/pdfdir/AlgoritmosProgramacion.pdf>
- Mendes, A. (may, 2006). ProLEARN, a platform to support programming. *Methods, Materials and Tools for Programming Education*, 1, 43-47.
- MIT. (2008). Scratch - Imagine. MIT. Recuperado de <http://wiki.scratch.mit.edu>
- MIT. (26 october, 2009). Faster Maintenance with Augmented Reality. *MIT Technology Review*. Recuperado de www.technologyreview.com/news/415977/faster-maintenance-with-augmented-reality/
- Monterrey, I. T. y de E. S. de. (2008). Aprendizaje Colaborativo, técnicas didácticas. *Programa de Desarrollo de Habilidades Docentes*. Recuperado de http://www.itesca.edu.mx/documentos7desarrollo_academico/metodo_aprendizaje_colaborativo.pdf
- Moser, R. (mayo, 1997). A fantasy adventure game as a learning environment: why learning to program is so difficult and hat can be done about it. *Paper presented at the Annual Joint Conference Integrating Technology into Computer Science Education*, Uppsala (Sweden).
- Murillo, M. (enero-abril, 2006). Exploring the teaching-learning process. *Actualidades investigativas en educacion explorando*, 6, 1–28.
- Myller, N. & Nuutinen, J. (enero-junio, 2006). JeCo : Combining Program Visualiz. *Informatic in Education*, 5 (2), 255-264.
- Oviedo, M. (2002). La enseñanza de la programación. *UPIICSA, México*. Recuperado de <http://bibliotecadigital.conevyt.org.mx/colecciones/documentos/somece2002/Grupo4/Oviedo.pdf>
- Redondo, M. (mayo, 2004). Aprendizaje en grupo de la programación mediante técnicas de colaboración distribuida en tiempo real. *Interacción*, 351–357.
- Suzuki, H. & Kato, H. (diciembre, 1997). Identity formation / transformation as the process of collaborative learning through AlgoArena. *CSC '97 Proceedings*, 280–289
- Szpiniak, A. & Rojo, G. (2007). Enseñanza de la programación. *Congreso de Tecnología en Educación y Educación en Tecnología, TE&ET'06*, La Plata, Buenos Aires (Argentina).
- Truong, N., Bancroft, P., & Roe, P. (junio, 2007) Web-Based Learning Environment: A Theory-Based Design Process for Development and Evaluation. *Journal of Information Technology Education*, 6, 23-43.
- University Carnegie Mellon. (2003). *Alice*. Recuperado de <http://alice.org/index.php>
- Vidal, C. (enero, 2014). Zerintia. Innovando con Madrid tour. *Revista de turismo y patrimonio cultural*, 12 (1), 239-241.
- Villalobos, J. (octubre, 2007). Proyecto Cupi2 – una solución integral al problema de enseñar y aprender a programar. *XXVII Reunión Nacional de Facultades de Ingeniería y VI Encuentro Iberoamericano de Instituciones de Enseñanza de la Ingeniería*, 10, 1–37.
- Vizcaíno, A., Contreras, J., Favela, J. & Prieto, M. (abril, 2002). An Adaptive, Collaborative Environment to Develop Good Habits in Programming. *Computer Science*, 1839, 262–271.