

## METODOLOGIAS ACTIVAS PARA LA ENSEÑANZA Y APRENDIZAJE DE ALGORITMOS BASADAS EN EL DESARROLLO DE VIDEO JUEGOS

Luis Norberto Zapata Puerta<sup>1</sup>  
Chaux<sup>2</sup>  
Área Programas Informáticos  
mas Informáticos  
Facultad de Ingenierías  
Politécnico Colombiano JIC  
lombiano JIC  
[lnzapata@elpoli.edu.co](mailto:lnzapata@elpoli.edu.co)<sup>1</sup>  
[poli.edu.co](mailto:poli.edu.co)<sup>2</sup>

Hernando Recaman  
Área Progra-  
Facultad de Ingenierías  
Politécnico Co-  
[hrecaman@el-](mailto:hrecaman@el-)

### Resumen

El objetivo de este artículo es compartir una estrategia metodológica que posibilita a los estudiantes adquirir un mayor y mejor aprendizaje significativo en el área de la lógica de programación, fundamentalmente en el desarrollo del pensamiento crítico y reflexivo, se centra primordialmente en la Enseñanza y Aprendizaje de la Lógica de Programación Basado en el Desarrollo de Video Juegos, dicha metodología busca desarrollar y potenciar algunas competencias tales como: trabajo en equipo y colaborativo, aprendizaje autónomo/independiente, mejor comprensión e implementación de los conceptos de las estructuras de control de los fundamentos de programación, mayor entendimiento del paradigma de Programación Orientada a Objetos (POO) y la construcción de artefactos de software mediante el lenguaje de programación, logrando así, una verdadera integración *Lógica+Paradigma POO+Lenguaje de Programación*, además de servir como agente motivador en un tema que apasiona y atrapa la atención de los estudiantes – Los video juegos.

**Palabras clave:** Metodologías activas, Juegos serios, enseñanza basada en juegos, Blended Learning, Modelos pedagógicos y TIC.

### Abstract

The objective of this article is to share a methodological strategy that enables students to acquire more and better significant learning in the area programming logic, mainly on the development of critical and reflective thinking, focuses primarily on teaching and learning programming logic based on the development of video games, this methodology seeks to develop and some competitions such as: teamwork and collaborative, autonomous/independent learning, better understanding and implementation of the concepts of structures for programming fundamentals, greater understanding of the paradigm of Object Oriented Programming (OOP) and building software artifacts using the programming language, thus achieving true integration *logic+OOP paradigm+programming language*, besides serving as a motivator in a topic that passion and catches the attention of students –the video games.

**Keywords:** active methodologies, serious games, game based learning, BLearning, pedagogical models and ICT.

### Introducción

Es sorprendente observar los altos niveles de cancelación y deserción estudiantil, y descenso en el número de matrículas de los programas de ingeniería de sistemas, ingeniería informática e ingeniería computacional no sólo del Politécnico Colombiano Jaime Isaza Cadavid, sino también de otras instituciones que enfrentan dicha problemática [1][2], y lo que es peor aún, mucho docentes seguimos ignorando sus causas. Es así como, los estudiantes de hoy han cambiado, no sólo su forma de vestir, sino también, su forma de hablar, bailar, pensar y comportarse, en general, su estilo de vida; estos cambios han sido influenciados de alguna forma por la cultura digital. La gran mayoría han pasado toda su vida rodeados y familiarizados con el uso de celula-

res, cámaras digitales, reproductores de música y videojuegos. “*El promedio de estudiantes universitarios de hoy dedica menos de 5.000 horas de su tiempo a la lectura, más de 10.000 horas atrapado en la diversión que brindan los videojuegos, además de las 20.000 horas frente a la televisión, es decir, que el Internet, los videojuegos, los celulares y el chat hacen parte integral de sus vidas*”, Prensky los llama *nativos digitales*, son hiperactivos, gastan parte de su tiempo sumergidos en los videojuegos, procesan información en paralelo, por eso, es frecuente observar a los estudiantes escuchar música mientras realizan otras tareas, además, en clase es difícil captar su atención por espacios de tiempo prolongados.

Por su parte, los profesores de hoy, gran parte de ellos inmigrantes digitales, *suelen tener poco aprecio por estas nuevas habilidades que los nativos han adquirido y perfeccionado a través de años de interacción y práctica, a estos, les gusta enseñar lentamente, paso a paso, una cosa a la vez, de forma individual, y sobre todo en serio*; Prensky afirma que es más factible, que los docentes *inmigrantes digitales* se adapten a los nuevos cambios, que los estudiantes “*nativos digitales*” retrocedan y aprendan las viejas costumbres. Existen algunos “acentos” para identificar a los “inmigrantes digitales”, tales como: guardar una dirección electrónica en una libreta en lugar de enviarla a “Mis Favoritos”, realizar una llamada telefónica para averiguar si recibió el correo electrónico, preferir modificar un texto en papel en lugar de editarlo en pantalla, preguntar cómo descargar archivos o videos de Internet, prefieren hacer largas filas en los bancos, en lugar de hacer los pagos por Internet, ya que desconfían que se realicen efectivamente las transacciones, Prensky (2001) [3].

Por lo anterior, es necesario diseñar e implementar propuestas didácticas orientadas a la creación de ambientes significativos de enseñanza y aprendizaje, con el empleo de metodologías activas que motiven a los estudiantes a aprender a aprender, apoyados en las nuevas tecnologías de la informática y las comunicaciones no como reemplazo, sino como apoyo permanente al proceso docente educativo. Nielsen (1995), introdujo el término de aprendizaje activo, a raíz de su trabajo con alumnos con discapacidades físicas; sostiene que el aprendizaje se manifiesta de forma participativa, a través de los juegos y la experimentación, es decir, se construye de forma activa más que pasiva [4].

Es así como se pretende fomentar el aprendizaje activo y significativo de la lógica de programación en los estudiantes mediante el desarrollo de videojuegos, propiciando poner a prueba los conocimientos adquiridos, evitando que la enseñanza de los fundamentos de programación se traduzcan en una simple memorización de conceptos, técnicas y estructuras sintácticas de algún lenguaje de programación, que a veces los alumnos deben adquirir por imitación del profesor; lo que se pretende aquí, es cambiar el enfoque y la forma de enseñanza y aprendizaje en esta área, donde el objetivo ya no será tan solo el de enseñar los conceptos básicos, las estructuras lógicas, y la utilización medianamente de algunos comandos del lenguaje de programación, por el contrario, va más allá, es desarrollar el pensamiento algorítmico, que potencie la capacidad analítica, el trabajo en equipo y colaborativo, la autonomía, el trabajo independiente, la curiosidad investigativa, y sobre todo, la creatividad; al desarrollar un videojuego se generará una serie de necesidades y retos que pondrán a prueba los conocimientos adquiridos y los que habrán de investigarse para aplicarlos de forma efectiva y significativa, por tanto, deberá integrar los conceptos, técnicas, estructuras lógicas, paradigma de programación (POO) y lenguaje de programación, en la solución del problema, lo que generará la incorporación de nuevos conocimientos a los ya adquiridos, de una forma divertida, ya que aprenderá “Jugando”. [10]

Para Ponsa, et al. [5] el aprendizaje activo se rige por los principios que tienden a favorecer el trabajo en equipo y colaborativo, aprendizaje autónomo, desarrollo de destrezas y habilidades, utilizar diversos medios, tecnologías y formas de representación de la información, propiciar el análisis, los debates y la crítica reflexiva, atender al ritmo individual de aprendizaje de cada alumno, utilizar distintas técnicas y procedimientos

de evaluación, combinar distintos tipos de estrategias y técnicas de enseñanza, las actividades prácticas, simulaciones o juegos de rol, análisis de casos, resolución de problemas y la elaboración de material didáctico de calidad o reutilizar otros ya creados.

**Aprendizaje basado en juegos:** Se inicia con los conceptos ofrecidos por las diferentes teorías de aprendizaje, entre las más importantes tenemos [6]:

1. La perspectiva conductista: Skinner. La secuencia de aprendizaje puede ser programada de acuerdo con las necesidades del alumno. Proporciona enseñanza individualizada que no requiere conocimientos previos.
2. aprendizaje significativo de Ausubel. El aprendizaje es significativo si es incorporado al conjunto de conocimientos del sujeto, relacionándolo con sus conocimientos previos. El contenido y la estructura de la asignatura son organizados por el profesor, el alumno se centra en la recepción.
3. Aprendizaje por descubrimiento de Bruner. Fundamentado en la resolución de problemas los cuales han de suponer un reto, un desafío que incite a su resolución y propicie la transferencia del aprendizaje.
4. La teoría de Piaget. El desarrollo de la inteligencia es una adaptación del individuo al medio. Los procesos básicos para su desarrollo son: adaptación (entrada de información) y organización (estructuración de la información).  
Constructivismo y mediación. Es posible que a través de la exploración individual el sujeto pueda adquirir determinados esquemas generales de conocimiento, pero mucho más difícil será que consiga alcanzar aprendizajes específicos.  
Será necesario definir la situación instructiva partiendo de las ideas previas de los sujetos, de sus intuiciones y también será preciso definir el tipo de intervención de otras personas: profesor y alumno.
5. Procesamiento de la información de Gagné. El fundamento básico para lograr el aprendizaje es conocer las condiciones internas que intervienen en el proceso y las condiciones externas que pueden favorecer el aprendizaje óptimo.

Los conceptos y teorías han influido directamente en la elaboración de esta propuesta metodológica que deseamos compartir con los lectores, y más específicamente en quienes buscan en las mejores prácticas una solución al problema de lograr una mejor enseñanza y aprendizaje de la lógica de programación. En la construcción de video juegos su principal forma de aprendizaje es a través de la experiencia de crear, escribir, diseñar y solucionar problemas. El juego computacional requiere de un excelente diseño teórico que piense en el tipo de alumno y su nivel, además de la recopilación de experiencias previas que valoren las posibilidades reales de producirlo y lo eficaz y divertido que puede ser. Es importante que el juego vaya acompañado de diferentes técnicas de motivación.

A continuación se muestran algunos aspectos que son mejorados por medio del juego [7]:

Desarrollo psico-motor	Desarrollo Cognitivo	Desarrollo Social	Desarrollo Emocional
Coordinación motriz	Estimula la atención, la memoria, la imaginación y la creatividad	Procesos de comunicación y cooperación con los demás	Desarrolla la subjetividad del niño
Equilibrio			Produce satisfacción emocional
Fuerza	Ayuda a diferenciar la fantasía y la realidad	Conocimiento del mundo	Controla la ansiedad
Manipulación de objetos	Fomenta el pensamiento científico y matemático	Preparación para la vida laboral	
Dominio de los sentidos			

Discriminación sensorial	Desarrolla el rendimiento	Estimulación del desarrollo moral	Controla la expresión simbólica de la agresividad
Coordinación visomotora	Estimula la comunicación, el lenguaje y el pensamiento abstracto	Favorecen la comunicación, la unión y la confianza en sí mismos	Facilita la resolución de conflictos
Capacidad de imitación		Potencia el desarrollo de las conductas prosociales	Facilita patrones de identificación sexual
		Disminuye las conductas agresivas y pasivas	
		Facilita la aceptación interracial	

Tabla 1. Aspectos mejorados por medio del juego

*Cada vez más los estudiantes demuestran gran interés y motivación por los video juegos y lo aceptan como una forma efectiva para aprender. Los juegos basados en experiencias, problemas y trabajo colaborativo forman un ambiente efectivo para el aprendizaje debido a sus características constructivistas; se logró determinar que existen tres factores motivantes en los juegos que son: el desafío mental, el reto físico y la experiencia social. [8]*

**La lúdica y el aprendizaje:** *La lúdica hace referencia a aquellas actividades diseñadas y programadas computacionalmente que son incluidas en el contenido programático de una asignatura. La correcta utilización de gráficos, animaciones, imágenes y los sonidos permiten alcanzar estos objetivos en todos los grupos educativos (Grupos que van desde la educación primaria hasta la educación Universitaria). Según O'Connor y Seymour la lúdica nos ofrece estimulación en la memoria, pues recordamos un **90% de aquello que hacemos**, un 10% de lo que leemos, un 20% de lo que oímos y un 30% de lo que vemos.*

*Teniendo en cuenta lo anterior, se observa que el principal método pedagógico que se enseña en los cursos de programación es "divide y conquistarás" utilizado con un nivel de informalidad que no pasa de ser una buena intención. Este enfoque, que se puede llamar aprendizaje por imitación, se basa en la hipótesis de que el estudiante es capaz de generar las habilidades necesarias tratando de imitar lo que el profesor hace. [9] En la enseñanza y aprendizaje de la lógica de programación se presenta una estructura basada en dos tipos básicos de actividades: actividades relacionadas con la teoría, en las que se adquieren los conceptos básicos relativos a la programación y actividades prácticas, en las que los alumnos deben transformar dichos conocimientos básicos en habilidades operativas, a través de su aplicación a variados problemas y situaciones. Para lograr buenos resultados se requiere la implementación de Metodologías Activas Constructivistas que generen un aprendizaje significativo y al mismo tiempo motiven al estudiante a aprender y divertirse construyendo juegos; ya que será éste quien debe desarrollar las habilidades mediante el descubrir por sí mismo, así como las técnicas para abordar los problemas de la programación. [10]*

*Además existen otros métodos pedagógicos para la enseñanza de la programación como:*

- 1. Entornos basados en ejemplos: son sistemas que realizan la búsqueda y selección de los ejemplos por medio de los siguientes criterios: selección conducida por el estudiante, selección conducida por el sistema y selección colaborativa (el sistema sugiere los ejemplos y el estudiante realiza la selección).*
- 2. Entornos basados en visualización y animación: este tipo de aplicaciones representa en forma esquemática los programas y algoritmos, de manera que a los alumnos les sea más fácil entenderlos.*

3. *Entornos basados en simulación: Facilitan el aprendizaje de la programación y la comprensión sobre el efecto que tiene la ejecución de instrucciones (codificadas en algún lenguaje de programación) mediante algún tipo de simulación. Los alumnos observan un “mundo imaginario” en el que habitan seres cuyo comportamiento viene dictado por la ejecución de las instrucciones del programa.[11]*

Lo fundamental en estos métodos pedagógicos es el uso de buenas estrategias didácticas que al ser implementadas en el computador permiten incrementar la participación creativa del alumno, además, que motiva a aprender. Por lo tanto, *si el estudiante no tiene una razón impuesta que le obligue a usar un determinado programa educativo, el sistema ha de ser capaz de atraer por sí mismo el interés del alumno creándole una motivación interna que le induzca a pasar las horas delante del computador y a dedicar más tiempo al autoaprendizaje. Este tipo de motivación es favorable para los alumnos aumenten la velocidad de aprendizaje, a la vez que incrementa el número de horas que invierte en usar el programa, vinculándolo en una espiral positiva que lo lleve a aprender cada vez más.[12]*

### **Herramientas que facilitan el aprendizaje de la programación.**

A continuación se hace un breve resumen de las herramientas más representativas para el aprendizaje de la programación:

Nombre	Descripción
Algolib	Facilita la creación de programas que realicen pruebas de algoritmos y toma medidas experimentales sobre el rendimiento del algoritmo. <b>Disponible en:</b> <a href="http://algo.inria.fr/libraries/">http://algo.inria.fr/libraries/</a> [10]
Alice	Nace como iniciativa para combatir la falta de interés en la programación, fué desarrollado por un equipo de Carnegie Mellon University; ofrece un entorno de programación en 3D, el cual permite crear animaciones y juegos interactivos de forma fácil y motivante, al mismo tiempo permite asimilar los conceptos de POO sin la necesidad de conocimientos previos. <b>Características principales:</b> No hay que recordar ninguna sintaxis especial. Utiliza un entorno sencillo de manejo basado en “arrastrar y soltar” para crear modelos en 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. <b>Disponible en:</b> <a href="http://www.alice.org.">http://www.alice.org.</a> [13]
C-jump	Los niños (mayores de 10 años) aprenden a programar lenguajes de computación como Java o C++ en una forma lúdica. Los jugadores (2 o 4) se convierten en esquiadores que deben bajar desde una montaña de la forma más rápida posible. Con cada descenso, los jugadores deben seguir las instrucciones que son similares a los códigos de los programas informáticos. <b>Disponible en:</b> <a href="http://c-jum.com">http://c-jum.com</a> [13]
Critical Mass	Se programa en C++ la eliminación de las piezas del juego pertenecientes al oponente. El juego de tablero es un entorno que fomenta la competencia interactiva que aumenta el esfuerzo y satisfacción del estudiante en la medida que completa las tareas. Fortalece el curso de estructura de datos. <b>Disponible en:</b> <a href="http://criticalmassgame.com/">http://criticalmassgame.com/</a> [14]
Cupi2	Este proyecto tiene como objetivo buscar nuevos métodos para enfrentar el problema de enseñar a programar, y al mismo tiempo que el estudiante desarrolle habilidades mediante el aprendizaje activo e incremental, con un enfoque pedagógico PBL (aprendizaje basado en problemas y proyectos). La nueva versión del curso incluye nuevos temas como XML, construcción de aplicaciones por componentes, desarrollo de pruebas sobre escenarios aleatorios y patrones de diseño (singleton, bridge, fábrica abstracta, iterador). Fué desarrollado por Jorge Alberto Villalobos Salcedo de la Universidad de los Andes – Bogotá. <b>Características principales:</b> Su modelo pedagógico se basa en el aprendizaje activo, el enfoque basado en problemas, aprendizaje incremental y aprendizaje por medio de ejemplos. Cuenta con tutoriales, laboratorios, frameworks, y entrenadores. <b>Disponible en:</b> <a href="http://cupi2.uniandes.edu.co/sitio/">http://cupi2.uniandes.edu.co/sitio/</a> [9]

eAdventure	<p>Es una herramienta para la creación de juegos educativos, en donde lo más importante es la narración de la historia, de fácil manejo y no requiere conocimientos previos de programación; permite hacer seguimiento al progreso del estudiante, exportar los juegos como objetos de aprendizaje, y capaz de ejecutar distintos contenidos. Desarrollado en la Universidad Complutense de Madrid.</p> <p><b>Características principales:</b> Se pueden crear dos tipos de juegos, en tercera persona en los que el jugador está representado por un avatar (ejemplo: para representar mundos fantásticos en los que el alumno aprende a la vez que avanza en la historia), y en primera persona (ejemplo: para la creación de "simulaciones tipo juego", en donde se simula de la forma más realista posible un entorno real). El autor del juego puede visualizar videos o presentaciones, incluir libros (especialmente útil para aquellos cursos en los cuales se emplea un texto guía).</p> <p><b>Disponible en:</b> <a href="http://e-ucm.es/">http://e-ucm.es/</a> [15]</p>
Game-Maker	<p>Es una herramienta que permite a los usuarios desarrollar videojuegos de forma fácil y sencilla sin tener que aprender un lenguaje de programación como Java o C++. Fue desarrollado por Markus Hendrik Overmars (Mark Overmars), profesor de programación de videojuegos en la Universidad de Utrecht, en los Países Bajos.</p> <p><b>Características principales:</b> la herramienta ofrece un sistema de eventos y acciones representados en íconos, para el desarrollo de videojuegos basta simplemente con "arrastrar y soltar", lo cual permite a los usuarios que no están familiarizados con la programación tradicional crear videojuegos intuitivamente. Contiene un lenguaje de programación de scripts llamado Game Maker Language (GML).</p> <p><b>disponible en:</b> <a href="http://www.comunidadgm.org/">http://www.comunidadgm.org/</a> [16]</p>
Jupiter	<p>Júpiter es una plataforma Web utilizada para la enseñanza y el aprendizaje de la programación avanzada de computadores, en la cual, los usuarios pueden probar sus habilidades en la creación de videojuegos de carácter competitivo, además, les permite involucrarse mucho más en el proceso aprendizaje, les da una perspectiva más avanzada, facilitando de esa forma un aprendizaje más profundo y significativo de la programación. Fue desarrollado en la Universidad Carlos III de Madrid.</p> <p><b>Características principales:</b> ofrece un escenario donde los usuarios (jugadores) pueden probar su código contra el resto de jugadores en un entorno de competición organizado en forma de liga a doble vuelta. Para ello ofrece a los estudiantes una Interface en la que el sistema y el código del jugador desarrollado por el usuario pueden comunicarse.</p> <p><b>Disponible en:</b> <a href="http://www.it.uc3m.es/rueda/Jupiter/">http://www.it.uc3m.es/rueda/Jupiter/</a> [10]</p>
Robocode	<p>Es un juego de programación, el objetivo es construir (programar) tanques de guerra para combatir contra otros tanques en un campo de batalla, utilizando el lenguaje de programación java o .Net. En este juego los jugadores escriben código para controlar un tanque robot miniatura, que deberá enfrentarse a otros similares que están siendo programados por otras personas. El objetivo del juego es construir el mejor tanque de guerra y destruir el resto; fue creado inicialmente por <b>Mathew A. Nelson</b> como un esfuerzo personal. IBM se interesó en Robocode como una forma divertida de empezar a aprender a programar en Java.</p> <p><b>Características principales:</b> Las principales actividades que se programan en el robot son: Moverse inteligentemente, esquivar ataques, apuntar con precisión y disparar adecuadamente.</p> <p><b>Disponible en:</b> <a href="http://robocode.sourceforge.net/">http://robocode.sourceforge.net/</a> [13]</p>
Scratch	<p>Es un lenguaje de programación que fué desarrollo por el grupo Kindergarten del Instituto Tecnológico de Massachusetts – MIT, está orientado principalmente a niños y jóvenes con el propósito de facilitar introducirse en el mundo de la programación de computadores; su interface ofrece la posibilidad de crear historietas interactivas, juegos, animaciones y simulaciones de forma fácil, atractiva y divertida.</p> <p><b>Características principales:</b> utiliza la técnica de apilar bloques de gráficos de instrucciones, es decir, que está basado en la técnica de pegar ladrillos de construcción tal como se utiliza en los legos, los bloques están prediseñados para encajar unos en otros sólo si son sintácticamente correctos.</p>

	<b>Disponible en:</b> <a href="http://scratch.mit.edu/">http://scratch.mit.edu/</a> [17]
Useractive	Los estudiantes son guiados a hacer sus propios descubrimientos (teoría constructivista). En cada fase del proceso se evalúa lo codificado para aprender sobre los errores, se continúa con la construcción de programas más complejos hasta que el estudiante domine el tema. El aprendizaje del lenguaje de programación se realiza online o a distancia. <b>Disponible en:</b> <a href="http://www.oreillyschool.com/2011/03/useractive-learning-starts-early-1/">http://www.oreillyschool.com/2011/03/useractive-learning-starts-early-1/</a> [18]

Tabla 2. Herramientas que facilitan el aprendizaje

### Experiencia en el Semestre 1 del Programa Técnico Profesional en Programación de Sistemas de Información del Politécnico Colombiano Jaime Isaza Cadavid.

Al igual que Gómez et al. (2009) [19], se comparte la idea fundamental de fijar las reglas de juego claras desde un principio, ya que siempre que se emprende un proceso innovador se corre el riesgo que los alumnos se sientan abrumados y pierdan la perspectiva sobre los objetivos que se deben alcanzar. En la siguiente tabla se puede visualizar la concertación de evaluación que se propuso en el módulo Desarrollo del Pensamiento Analítico y Sistemático I equivalente a la asignatura Fundamentos de Programación 1 que se imparte en el primer semestre de dicho programa:

N	Temática	Actividades	%
1.	Conceptos básicos de algoritmos y Ejercicios de lógica cotidiana	- Consulta - Taller	10
2.	Solución de problemas implementado condicionales con subprogramas	- Consulta - Taller - Práctica de laboratorio	10
3.	Solución de problemas implementado ciclos con subprogramas	- Consulta - Taller - Práctica de laboratorio	10
5.	Solución de problemas implementado Arreglos con subprogramas	- Consulta - Taller - Práctica de laboratorio	10
6.	Solución de problemas implementado Arreglos con POO	- Consulta - Taller - Práctica de laboratorio	10
7.	Desarrollo de un Video Juego implementando ALICE / proyecto empresarial	- Socialización y evaluación del Video Juego / proyecto empresarial	20
8.	Primera y segunda entrega del Proyecto Pedagógico Integrador (PPI)	- Entregables y Socialización	30
		<b>TOTAL</b>	<b>100</b>

Tabla 3. Concertación de evaluación

Como se puede observar en la tabla anterior, se introdujo varios aspectos innovadores para el desarrollo del curso, los cuales han permitido obtener una mayor motivación y rendimiento académico por parte de los estudiantes, estos son:

a. *Aprendizaje Basado en el Desarrollo de Video Juegos.* Con esta metodología se pretende motivar a los estudiantes a aprender a programar por medio del desarrollo de video juegos, tomando como base, que la gran mayoría de estudiantes suelen invertir gran parte de su tiempo en los juegos electrónicos por el alto grado de interés que despierta en ellos. Por tal motivo, se decidió utilizar la herramienta Alice (<http://www.alice.org>), que tiene un entorno de programación el cual utiliza gráficos y objetos en 3D, el desarrollo resulta ser una tarea simple y rápida, ya que oculta la complejidad de la construcción de estructuras lógicas [20][21].



## Ilustración 1. Interfaz Alice

b. *Aprendizaje Basado en Proyectos*. La experiencia de enseñar a programar por más de 20 años, ha demostrado que es fundamental tener en cuenta los intereses y pasiones de los estudiantes como parte del desarrollo del curso, para lograr captar su atención y sumergirlo de forma gradual en las temáticas de clase; es así como, al iniciar el curso se propone a los estudiantes seleccionar un tema de investigación, como por ejemplo: Flash, ActionScript, HTML, Javascript, Php, Java, C++ , OpenGL, Illustrator, entre otros, y trate de desarrollar una solución, es decir, un programa desde la herramienta elegida, que luego al final del curso debe socializar con los demás compañeros del grupo, esto permite entre otras cosas, que el estudiante desarrolle varias competencias, como: aprendizaje autónomo y trabajo en equipo.

c. *Aprender Haciendo*. Parte de la metodología del curso consiste en crear la “Necesidad” para que el aprendizaje sea significativo, es decir, hacer que el estudiante se involucre con un tema que le interese y le apasione, como (Flash, ActionScript,...), y de esta forma lograr que pase mayor tiempo consultando en Internet o realizando actividades en su PC; así se podrá generar la disciplina del autoestudio y poco a poco se irá involucrando en las temáticas de clase atraído, como si fuera una golosina o “carnada” para atraer su atención, una vez logrado este primer paso, el estudiante podrá llegar a clase con lecturas previas de un tema en particular, antes que el profesor inicie su exposición magistral; se sabe por experiencia lo bien que se aprende cuando debemos exponer o socializar un trabajo ante el grupo. Complementando lo anterior, por cada capítulo los alumnos deben realizar una práctica del taller que se propone, utilizando el lenguaje de programación Java/C++; a medida que se desarrollan los temas magistrales (contenido de la guía didáctica) se van dando las pautas para que los estudiantes realicen las prácticas y proyectos (video juegos o empresarial).

Se conformaron 8 equipos, cada uno compuesto por 3 integrantes, los primeros 4 equipos emplearon los conocimientos adquiridos en Java/C++ para desarrollar un sistema de información empresarial. Los siguientes 4 equipos utilizaron la herramienta de desarrollo *Alice* para crear un video juego que partiera de su propia iniciativa.

Los primeros 4 equipos desarrollaron varios sistemas de información, tales como: sistema de inventarios para supermercados, manejo de citas para consultorios odontológicos, alquiler de películas para tiendas de video y servicios de centros veterinarios. Los otros 4 equipos trabajaron en el desarrollo de los siguientes video juegos: rescate de rehenes desde helicópteros, guerra entre naves espaciales, carreras de autos y tiro al blanco.

Se pudo observar que los estudiantes que desarrollaban sistemas de información para empresas dedicaban menos tiempo al desarrollo del proyecto, además, presentaban dificultades para desarrollar las actividades de trabajo en equipo y trabajo independiente, por otro lado, consideraban las tareas del proyecto como tediosas y aburridas, ya que algunos de ellos manifestaban abiertamente no tener interés en aprender sobre contabilidad, inventarios, entre otros; al mismo tiempo, consideraban las actividades de los otros equipos como más divertidas, y por ende, la evaluación de su desempeño al final del periodo académico fue muy baja, se pudo evidenciar su bajo desempeño y desmotivación para hacer entrega de los artefactos funcionales. Por su parte, los equipos que diseñaron los video juegos, presentaron algunos inconvenientes en la etapa inicial en cuanto al manejo de la herramienta *Alice* (manejo del IDE), pero una vez sobrepasada esta dificultad, el desarrollo resultó en una tarea divertida y emocionante,

logrando que el tiempo de dedicación por parte del equipo y de cada integrante fuera muy alto, especialmente por la gratificación de ver crecer su video juego en cada una de las etapas del proyecto, obtuvieron mejores resultados cuantitativos sobre los conocimientos adquiridos en lógica, estructuras de control y Programación Orientada a Objetos.

### **Conclusiones y trabajo futuro**

Uno de los principales inconvenientes que se presentan en el desarrollo de los video juegos es la creación de contenidos. Alta demanda de tiempo que requiere el diseño estructural del juego, la complejidad en la creación de determinadas escenas, guiones, objetos, actores, roles de los personajes y niveles.

El diseño de los cursos de fundamentos de programación y en especial la enseñanza de la lógica de programación, a través del desarrollo de video juegos deben ser minuciosamente planeados, ya que se pueden desvirtuar los objetivos que se pretenden alcanzar, es decir, que en lugar de aprender a programar y conocer las complejidades del código, por el contrario, se limite sólo al diseño visual (diseño gráfico), además, de la ocultación del código y la creación de animaciones.

Existe actualmente, la necesidad de seguir investigando sobre las nuevas metodologías de enseñanza y aprendizaje y demás técnicas que permitan “despertar” la pasión de los estudiantes por la lógica de programación, como también, poder involucrar sus intereses personales dentro de las temáticas que se desarrollan en clase, permitiendo que formen parte activa en la construcción de su propio conocimiento y no como simples receptores pasivos que se limitan a transcribir información de un medio a otro, como por ejemplo, del tablero al cuaderno o de Internet a la tableta digital.

A partir de la experiencia se puede percibir un futuro promisorio, puesto que algunos estudiantes han comenzado a reflejar y dar indicios que indican que el trabajo realizado no ha sido en vano. Al respecto conviene mencionar los aspectos de mayor relevancia:

- La disciplina y el interés que se ha reflejado en la elaboración de los proyectos
- La actitud y el compromiso asumido frente a la socialización grupal
- La capacidad de autoaprendizaje que han alcanzado
- El desarrollo de la capacidad de trabajo en equipo

### **Bibliografía**

- [1] Prensky M., Nativos Digitales, *Inmigrantes Digitales*, MCB University Press, Vol. 9 No. 5, Octubre 2001.
- [2] Leutenegger S., Edgington J., “A Games First Approach to Teaching Introductory Programming”. Proceedings of the 38th SIGCSE technical symposium on Computer science education. Vol. 39 Issue 1. Nueva York. 2007.
- [3] Prensky, M. Digital Game-based Learning. McGraw-Hill. 2001.
- [4] López, C. “Propuesta de un programa de estrategias metodológicas para el desarrollo de habilidades básicas en escolares de primer ciclo en cuba”. Disponible en <http://hera.ugr.es/tesisugr/16554255.pdf>. [Consultado el 16 de mayo de 2009]. 2007.
- [5] Ponsa, P., Román, J., Díaz, M., Ferran, J., “Propuesta de herramientas para la introducción de metodologías activas en el proceso de enseñanza/aprendizaje”. Disponible en <http://www.epsevg.upc.edu/fdv/docs/doc1044.pdf>. [Consultado el 16 de mayo de 2009]. 2005.
- [6] Rogoof, B. *Aprendices del pensamiento. El desarrollo cognitivo en el contexto social*. Barcelona, Paidós. 1993.
- [7] Ramírez, Santos Urbina. *Informática y Teorías de Aprendizaje*. 2003.
- [8] Whitton, N., *Motivation and computer game based Learning*, 2007.
- [9] Villalobos, J., “Proyecto CUIP2 – una solución integral al problema de enseñar y aprender a programar”, 2009.
- [10] Chicharro, M., García, J., Ramírez, R., “Júpiter: Un entorno web para el aprendizaje basado en juegos competitivos”, 2008.
- [11] Gómez, M., “Una revisión de métodos pedagógicos innovadores para la enseñanza de la programación”, 2005.

- [12] Gómez, M., Gómez, P., González, P., "Aprendizaje Basado en Juegos". Universidad Complutense de Madrid, España, 2005.
- [13] Gómez, M. "Una revisión de métodos pedagógicos innovadores para la enseñanza de la programación". Disponible en [http://bioinfo.uib.es/~joemiro/semDOC/PlanEstu-dis/JENUI/RevisiónMetodosPedagogicosInnovadoresProgramacion\\_jenui2003.pdf](http://bioinfo.uib.es/~joemiro/semDOC/PlanEstu-dis/JENUI/RevisiónMetodosPedagogicosInnovadoresProgramacion_jenui2003.pdf). [Consultado el 16 de mayo de 2009]. 2003.
- [14] Lawrence, R., "Teaching Data Structures Using Competitive Games", IEEE Transactions on Education, 2004.
- [15] Grupo de investigación e-UCM, "<e-Adventure> proyecto de sistemas informáticos", Universidad Complutense de Madrid, 2007.
- [16] Habgood, J., Overmars, M., "the Game Maker's Apprentice: Game Development for Beginners", Apress, 2006.
- [17] Resnick, M., "Scratch: programming for all", communications of the ACM, Vol. 52, num. 11, noviembre de 2009.
- [18] O'reilly school of technology, O'reilly media, catalogo junio-diciembre, 2012.
- [19] Gómez, E., García, M., Villalba, G., "Aplicación de diversas metodologías activas en la asignatura de introducción a la programación". Universidad Europea de Madrid. Disponible en <http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2005/gomapli.pdf> [Consultado el 3 de mayo de 2009]. 2009.
- [20] <http://www.alice.org>, Carnegie Mellon University, [Consultado 9 de febrero de 2013]. 2013.
- [21] Moskal, B., Lurie, D. y Cooper, S., "Evaluating the Effectiveness of a New Instructional Approach". Pensilvania: Carnegie Mellon University. 5 p. Disponible en <http://www.alice.org/publications/pubs/EvaluatingTheEffectivenessOfANewApproach.pdf> [Consultado el 8 de diciembre de 2009]. 2006.

# SOCIEDAD DE LA INFORMACION

[www.sociedadelainformacion.com](http://www.sociedadelainformacion.com)

Edita:



Director: José Ángel Ruiz Felipe  
Jefe de publicaciones: Antero Soria Lu-  
ján

D.L.: AB 293-2001

ISSN: 1578-326x