



Innovation in research and engineering education:
key factors for global competitiveness

*Innovación en investigación y educación en ingeniería:
factores claves para la competitividad global*

UN MODELO PARA LA CREACIÓN DE ESCENARIOS DE APRENDIZAJE EN INFORMÁTICA

Juan Albornoz, Raúl Chaparro, María Díaz

Escuela Colombiana de Ingeniería Julio Garavito
Bogotá, Colombia

Resumen

En informática es indispensable la apropiación de conceptos y estrategias abstractas que al aplicarse a la solución de problemas conducen a resultados correctos y eficientes. Para lograr un aprendizaje más efectivo, lúdico y natural de los mismos se propuso diseñar experiencias especiales de aula. Los resultados son más de cincuenta escenarios probados y reutilizables y, lo más importante, un modelo marco para orientar al docente sobre su uso y creación.

Palabras clave: aprendizaje; computación fundamental; computación bioinspirada

Abstract

Computer Science requires the acquisition of abstract concepts and strategies that when applied to problem solving lead to correct and efficient results. To achieve more effective, playful and natural learning of these a program was formulated to design special classroom learning experience. The results are more than fifty scenarios tested and reusable and, most importantly, a model framework to guide teachers in their use and creation.

Keywords: learning; fundamental computing; bio-inspirate computing

1. Introducción

Escenarios de Aprendizaje es un programa de investigación de la Decanatura de Ingeniería de Sistemas de la Escuela Colombiana de Ingeniería Julio Garavito (ESCUELA) que tiene como propósito desarrollar y orientar el desarrollo de experiencias de aprendizaje propicias para lograr la comprensión de conceptos y principios, la apropiación de estrategias de solución de problemas y el desarrollo de habilidades de investigación en el área de la informática. (Albornoz & Chaparro, 2002)

La evolución de este modelo pedagógico basado en escenarios ha estado fundamentada en los lineamientos de formación de la institución y en la búsqueda continua de una mayor comprensión integral de tres aspectos esenciales para el aprendizaje: el saber que se va a aprender, el perfil de quienes van a aprender y la forma como se aprende, sustentada en el estudio de teorías relevantes y el desarrollo y evaluación de los resultados obtenidos durante estos diez años.

El constructivismo social, que afirma que el conocimiento debe ser descubierto y estructurado por los estudiantes en comunidad, es la teoría de aprendizaje central de esta propuesta inspirada especialmente en ideas de Ausubel, Vigotsky, Piaget, Lakoff y Radford. El área de interés de este trabajo es la solución de problemas en informática considerando dos aproximaciones a la programación: la clásica fundamental y la moderna bioinspirada.

El resultado principal de este programa es un modelo marco para el desarrollo de lo que se han llamado **escenarios de aprendizaje**, que decanta el conocimiento y la experiencia logrados en la construcción, aplicación y evaluación de más de cincuenta propuestas de experiencias en aula para que pueda ser aprovechada por otros docentes que deseen desarrollar nuevas propuestas de aprendizaje en sus áreas de conocimiento.

Este artículo se inicia con una descripción del modelo marco que incluye las teorías que lo sustentan y la estructura propuesta para aprendizaje; se continúa con la descripción de dos experiencias exitosas, una en el área de computación fundamental y otra en computación bioinspirada, y por último, se presentan conclusiones.

2. Fundamentos del modelo marco

La ESCUELA está comprometida con la formación de profesionales que estén “en condiciones de formular soluciones de carácter autóctono a problemas nacionales y llevar a feliz término tales soluciones” mediante una formación integral formulada como “un proceso solidario de estudiantes y profesores motivados por la comunidad de objetivos, por lo cual se requiere la participación activa de unos y otros en dicho proceso”(ESCUELA, 2002). Estos lineamientos orientaron la exploración de las teorías de aprendizaje que dan sustento teórico al **modelo marco de aprendizaje** propuesto.

La idea básica de la propuesta es de Ausubel, quien sintetiza su aporte: “El factor más importante que influye en el aprendizaje es lo que el alumno ya sabe. Averígüese esto y enséñese consecuentemente” (Ausubel et al, 1976), que motivó la búsqueda de un elemento que permitiera efectivamente acercarse al mundo del estudiante. Este elemento, **la metáfora**, se encontró elaborado en los trabajos de Lakoff: “Hemos llegado a la conclusión de que la metáfora impregna la vida cotidiana, no solamente el lenguaje, sino también el pensamiento y la acción” (Lakoff & Johnson, 1980).

Para poder trabajar con la metáfora, que al igual que la teoría es abstracta, el siguiente paso fue diseñar herramientas para concretizarlas, llamadas **modelos para aprendizaje**, considerando los distintos niveles de conocimiento y la riqueza de medios para representarlo. Radford, en su teoría del aprendizaje, expone “la importancia de experimentar con todo el cuerpo”.(Radford, 2013). Los modelos se diseñaron aprovechando diferentes medios para involucrar todos los sentidos en el proceso de aprendizaje y lograr que el estudiante se mueva con fluidez de lo práctico a lo teórico y de lo concreto a lo abstracto.

Los modelos para aprendizaje, además de concretizar la metáfora y la teoría, permiten propiciar actividades con base en guiones llamados **escenarios**, en las que equipos de estudiantes deben afrontar y resolver un reto. En estas actividades ellos hacen conjeturas, experimentan, discuten resultados, proponen ideas de solución, construyen soluciones, integran soluciones parciales, etc. Los escenarios pretenden, tal como indica Piaget, “enfrentar al estudiante a situaciones en las que experimente en el más amplio sentido de la palabra: probar cosas para ver qué pasa, manejar objetos, manejar símbolos, plantear interrogantes, buscar sus propias respuestas” (Piaget, 1972) y vivencien el proceso de aprendizaje descrito por Vigotsky: “un proceso interpersonal se transforma en un proceso intrapersonal. Cada función aparece dos veces en el desarrollo cultural de la persona: primero, a nivel social, y luego a nivel individual” (Vygotsky, 1995)

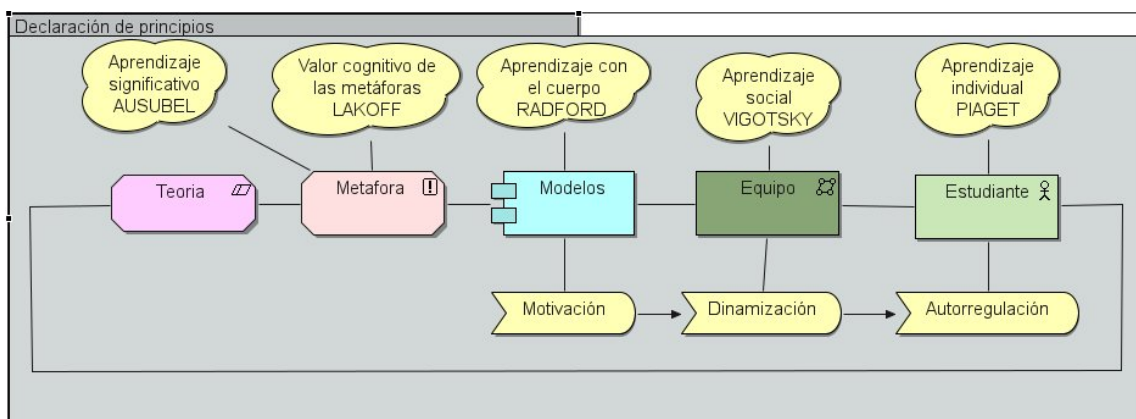


Figura 1. Elementos y teorías

Con el esquema anterior (figura 1) se busca responder al modelo de formación colaborativo y activo propuesto por la ESCUELA, en su compromiso con profesionales que logren cultivar sus competencias para formular opciones de solución y llevar a feliz término la mejor.

3. Descripción modelo marco

La estrategia de aprendizaje consiste en la selección de una **metáfora** adecuada que represente los elementos esenciales de la **teoría** de estudio; la construcción de variadas herramientas, llamadas **modelos para el aprendizaje**, que concretan la metáfora y la teoría; y finalmente, el diseño de **escenarios de aprendizaje** planteando un reto desafiante y preparando el ambiente adecuado para su solución por parte de un **equipo de trabajo** (figura 2.a).

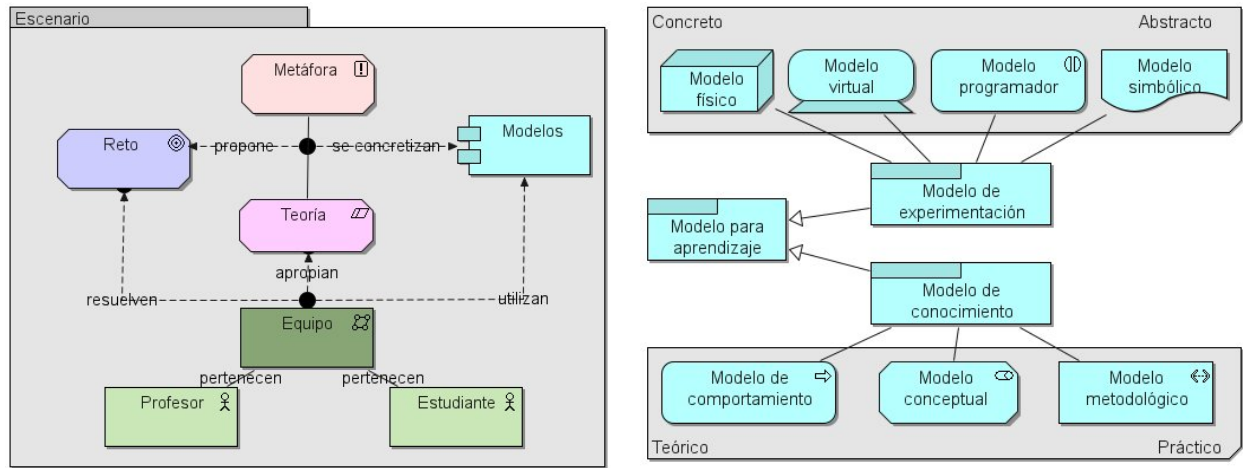


Figura 2. (a) Esquema de escenario (b) Clasificación de modelos

A. Teoría

En primer lugar se deben identificar claramente los conceptos y estrategias de la teoría informática por trabajar. Aquí es importante tanto la experiencia docente como el dominio de la teoría subyacente. Para esto se recomienda el uso de mapas conceptuales que servirán como semilla para los modelos de conocimiento.

B. Metáfora

La selección de la metáfora apropiada es un punto crítico en esta propuesta. Para que sea efectiva debe representar los elementos de la teoría, permitir generar diversos retos y ser familiar e interesante para los estudiantes. La metáfora ofrece el lenguaje para que profesores y estudiantes puedan expresar y compartir sus preguntas e ideas sobre la nueva área de conocimiento, desde el inicio del proceso de formación, y facilitar la apropiación del lenguaje informático correspondiente, necesario en etapas posteriores.

C. Modelos

Los modelos que se proponen son de dos clases complementarias: de conocimiento y de experimentación (figura 2.b). Los modelos de conocimiento sirven como mapas visuales que orientan permanentemente el recorrido de los estudiantes por el nuevo territorio conceptual; y los modelos de experimentación permiten que los equipos de trabajo interactúen para apropiarse y resolver el problema.

Los modelos de conocimiento ilustran de manera sintética el conocimiento relevante tanto declarativo como procedimental del área respectiva y su analogía en la metáfora, de tal forma que se pueda comprender el comportamiento en estos dos referentes. Los modelos propuestos van de lo teórico a lo práctico: conceptual, de comportamiento y metodológico.

Los modelos de experimentación son artefactos físicos, *software* o lógicos que permiten experimentar con la metáfora para proponer problemas, hacer conjeturas y visualizar, construir y evaluar posibles opciones de solución. Estos modelos van de lo concreto a lo abstracto: físico, virtual, del programador y simbólico.

D. Escenarios

En el proceso de aprendizaje propuesto, la metáfora aparece como el primer contacto del estudiante con el nuevo conocimiento; así, los escenarios iniciales deben permitirle al estudiante familiarizarse con la

metáfora, al profesor, indagar lo que saben los estudiantes y a ambos, generar un ambiente de motivación y confianza. Motivación, al hacerse conscientes de las posibilidades que abren los nuevos conocimientos y confianza en que en equipo van a superar los retos y, por lo tanto, aprender.

Después de exponer al estudiante a los escenarios iniciales, se proponen desafíos cada vez más complejos para que apropien con más profundidad conceptos y estrategias. Finalmente, se exponen a escenarios en los cuales para resolver los retos tienen que cambiar de representación usando el lenguaje de la teoría y herramientas de desarrollo profesionales. Esta última etapa permite completar el proceso de aprendizaje.

4. Computación fundamental. A la caza de invariantes

En esta línea se busca que los estudiantes apropien los conceptos y estrategias básicas de la computación, como son: problema bien definido, algoritmo, estado, precondition, poscondición, cambio de estado, invariante, recurrencia, corrección y eficiencia.

En este trabajo se ha encontrado que la máquina de Turing es tal vez el modelo más sencillo de entender que encierra todo el poder de la computación; sobre ella se pueden introducir los conceptos claves de programación y desarrollar hábitos y metodologías de trabajo (Hodges, 1995). Sin embargo, se debe presentar de manera efectiva a los estudiantes. Para esto se han desarrollado modelos de experimentación que permiten la interacción empleando los sentidos (táctil, visual y auditivo); gracias a ellos, se ha rescatado la máquina de Turing como un instrumento didáctico para introducir la programación de manera muy sencilla, sin necesidad de abordar lenguajes simbólicos ni herramientas tecnológicas complejas. Los estudiantes programan inicialmente con el modelo físico y luego pasan, de manera natural, al modelo virtual, apropiado para retos más complejos (Albornoz & Chaparro, 2005).

Así mismo, se ha identificado que los juegos discretos son como una fuente inagotable de posibilidades para plantear metáforas y retos motivantes y naturales dentro de la computación fundamental. A modo de ejemplo, se ilustrará la propuesta de uso de un juego discreto popular, **las torres de Hanoi**, para enseñar una de las estrategias claves para resolver un problema de naturaleza algorítmica: encontrar un invariante que capture la esencia de la solución. Esta misma metáfora resulta útil para otros conceptos fundamentales como son la explosión en tiempo de las soluciones exponenciales, y problemas de naturaleza recurrente.

A. Teoría y metáfora

Un invariante es un patrón o propiedad que se repite a lo largo de todo un proceso. Si el invariante caracteriza suficientemente la esencia de la solución, se dice que es un invariante fuerte, y por tanto, el seguirlo conducirá al algoritmo de solución. Las habilidades necesarias para formular invariantes son críticas en la computación fundamental para derivar soluciones correctas.

En este caso la metáfora viene implícita en el juego y sus reglas, lo cual se puede enunciar así: en una antigua ciudad de la India, según cuenta la leyenda, los monjes de un templo tenían que mover una pila de 64 discos sagrados, hechos en oro, de una torre de diamante a otra. Los discos eran tan pesados que sólo se podía cargar uno a la vez. Un disco no podía ser puesto sobre uno más pequeño y sólo había una torre intermedia entre el lugar de origen y el de destino, donde se podía descargar el disco para descansar. La leyenda dice que, antes de que los monjes hicieran el último movimiento para dejar todos los discos en su destino, el templo se volvería polvo y el mundo se acabaría.

B. Modelos

Para comprobar esta teoría se han construido modelos físicos de tablero, aptos para interactuar y discutir en clase entre todos, y una diversidad de modelos de trabajo en equipo con diferentes versiones: la clásica, para explorar inicialmente el comportamiento; la circular (figura 3 a.), que permite detectar algunos invariantes rotativos, y la de dos colores (figura 3 b.), que ha permitido encontrar un invariante novedoso. Además, usualmente los estudiantes motivados consiguen un juego ya fabricado para experimentar.



Figura 3. (a) Modelo físico circular (b) Modelo físico dos colores

Como modelo virtual, se cuenta con un programa desarrollado para simular múltiples juegos discretos, diseñado con características especiales de usabilidad como sencillez, transparencia y controlabilidad, que ofrece opciones adicionales que permiten reintentar y almacenar e imprimir los intentos de solución para el análisis de los mismos.

C. Escenarios

El propósito del primer escenario es que los estudiantes se familiaricen con el juego y el cumplimiento de las reglas, primeros invariantes. En un segundo escenario, el reto es resolver el problema con pocos discos: 1, 2, 3, 4, 5, encontrando un invariante o condición que se pueda expresar de manera clara, que sin ambigüedad determine cada paso y que conduzca a la solución óptima para cualquier número de discos.

En un tercer escenario, orientado a análisis de complejidad, el reto es encontrar una fórmula para calcular el número de pasos necesarios para solucionar el problema en función del número de discos llenando una tabla para estimar el tiempo necesario suponiendo que se tarda un segundo en mover cada disco, o un milisegundo, o un microsegundo, etc. Finalmente, deben justificar por qué su fórmula es correcta.

En un escenario final se les pide implementar en el computador su algoritmo de solución.

D. Resultados

- Con los modelos de experimentación, los estudiantes logran resultados más rápidamente que cuando se planteaban este problema, que es clásico, sin que mediara el modelo. De esta manera, se ha podido asumir retos y preguntas más complejas.
- Se puede abordar y discutir los temas de la “recurrencia” e “intratabilidad” en etapas tempranas. Los

estudiantes detectan la naturaleza recurrente del problema y palpan la explosión temporal de soluciones exponenciales.

- El modelo que tiene los discos de dos colores alternados facilitó descubrir un invariante novedoso que cualquiera puede seguir, y trivializa completamente el juego: “En cada paso, asegúrese de no devolverse al estado anterior, y que no queden dos discos del mismo color ni vertical ni horizontalmente”.

5. Computación bioinspirada. Cúmulos de partículas

La admiración que produce observar algunos comportamientos naturales -como a una bandada de aves migrando o a un cardumen de peces que se protegen de un depredador- ha motivado estudios para comprenderlos; sin embargo, sólo gracias al desarrollo de la informática los biólogos han logrado ir develando sus misterios (Garnier, 2007). De ese trabajo interdisciplinario surge la **programación bioinspirada**, al descubrirse que simulando estos comportamientos se pueden obtener soluciones a problemas informáticos difíciles con simplicidad y eficacia, como lo hace la naturaleza. A pesar de esto, no se ha incorporado en la formación de pregrado como alternativa disponible para solución de problemas.

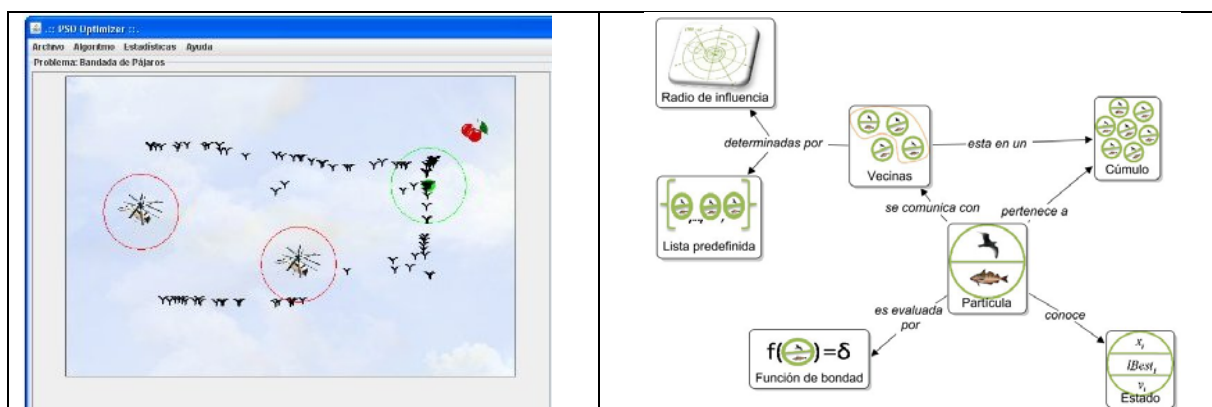
En 2006 se decidió usar el **marco modelo de escenarios** para propiciar el aprendizaje de tendencias bioinspiradas y hasta la fecha se han trabajado seis teorías con buenos resultados (Díaz et al, 2011). Para ilustrar esta línea de trabajo, se presenta la solución de problemas de optimización con un algoritmo inspirado en el desplazamiento social de los enjambres: **Optimización por Cúmulo de Partículas** (PSO *Particle Swarm Optimization*) (Casallas & Vargas, 2009).

A. Teoría y metáfora

En la programación bioinspirada la metáfora no debe inventarse porque se basa en comportamientos de seres de la naturaleza que, en menor o mayor grado, son familiares a los estudiantes y que si se les guía a observarlos llegan a interesarles profundamente. La metáfora en **PSO** es, por tanto, el **enjambre**: una agrupación considerable de animales de una misma especie que persiguen un objetivo de interés común y que lo logran gracias a la cooperación de sus individuos realizando acciones muy simples (Poli et al, 2007)

B. Modelos

El **primer paso**, quizás el más importante, consiste en hacer visible el misterio y la intencionalidad del comportamiento de los enjambres para que el estudiante busque y logre comprenderlo. Con este propósito se desarrolló un modelo virtual que le permite explorar el comportamiento del enjambre y los modelos conceptual y de comportamiento como medios que facilitan entenderlo.



En el **modelo virtual** se simula el comportamiento de una bandada de pájaros que buscan alimento (objetivo), en un espacio con helicópteros (restricciones) usando el algoritmo PSO. El estudiante puede apreciar cómo al modificar la posición del alimento los pájaros convergen hacia éste evadiendo la posición de los helicópteros. Adicionalmente, pueden experimentar con diferentes valores en los parámetros estándar de PSO (número de partículas, velocidad y tipo de enjambre, etc.) visualizando el cambio en el comportamiento de la bandada (figura 4 a.). El **modelo conceptual** está formado por dos mapas conceptuales con vistas alternativas del mismo fenómeno: el primero con la biológica de los enjambres y el segundo con la informática de los cúmulos de partículas (figura 4 b.). La idea es que una vez el estudiante comprenda el comportamiento de un enjambre transfiera esta comprensión a los elementos informáticos que los representan. El **modelo comportamiento** se desarrolló para ilustrar un detalle crítico de comportamiento en los enjambres: la toma de decisión de los individuos sobre el cambio en su movimiento que es crítica para construcción de una solución informática de este tipo.

El **segundo paso** es lograr que los estudiantes puedan resolver problemas informáticos utilizando PSO. Para esto se desarrollaron dos modelos, el metodológico y el del programador.

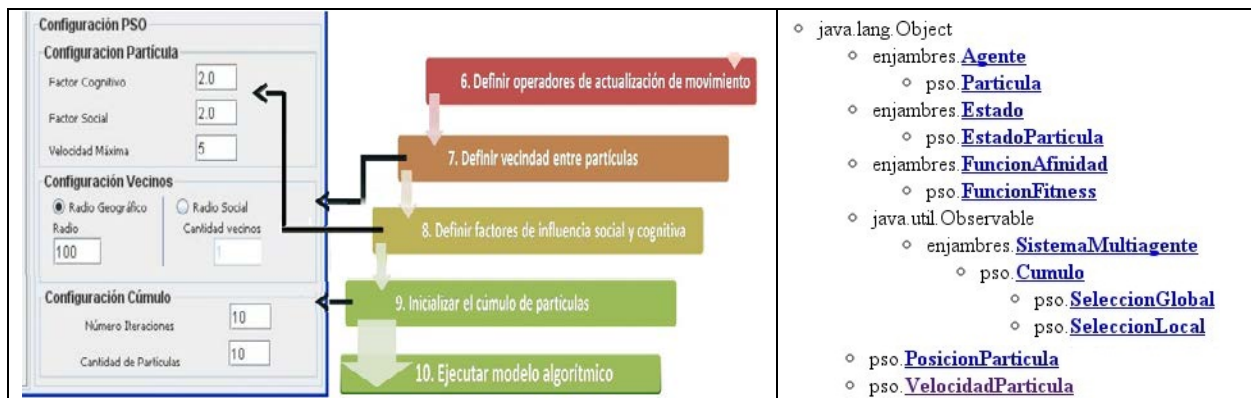


Figura 4 (a) Modelos del programador y metodológico (b) Modelo del programador

El **modelo metodológico** guía al estudiante para que a partir de la descripción de un problema llegue a una buena solución en tres fases: formular el problema en términos de optimización, enunciarlo para que sea resuelto por un cúmulo de partículas, y programar y sintonizar la solución en el modelo del programador (figura 4 a.). El **modelo del programador** es un *framework* en lenguaje java para que los estudiantes construyan fácilmente su solución. En la capa de aplicación se tienen construidos como componentes *software* la estructura planteada en la vista informática del **modelo conceptual** (figura 4 b.). La capa de presentación ofrece controles para sintonizar los parámetros de PSO y visualizar el comportamiento.

C. Escenarios

En esta tendencia se tienen desarrollados tres escenarios, además de los iniciales, con retos de diversa naturaleza y dificultad. El primero, de dificultad baja, es el juego **Fiver**, que consiste en colorear un tablero de juego en la menor cantidad de pasos; el segundo, de dificultad media, es el clásico **problema del viajero** en el que se busca una ruta óptima para recorrer un conjunto de ciudades de una región; y el último, de dificultad alta, es un problema real de **asignación** de proyectos de grado a estudiantes maximizando la satisfacción de éstos y los directores. Tales escenarios marcan hitos en un camino en el que el estudiante

adquiere y afianza las competencias necesarias para solucionar problemas usando esta tendencia.

D. Resultados

- El nivel de entendimiento del PSO fue más profundo en los estudiantes que se acercaron a él usando los modelos de conocimientos desarrollados que leyendo la literatura técnica.
- Las soluciones planteadas por los estudiantes a los tres retos propuestos les han permitido experimentar la riqueza, simplicidad y eficacia de esta aproximación.
- Los estudiantes han logrado comprender la naturaleza e implicaciones de los sistemas complejos al estudiar sistemas naturales análogos como los enjambres.

5. Conclusiones

1. El Marco Modelo basado en Escenarios es una guía flexible muy útil en diversas áreas de la informática, y que se podría aprovechar en otros campos de conocimiento.
2. Las metáforas encontradas les han permitido a los estudiantes entrar en el mundo del problema de manera rápida y motivada y enfocar rápidamente su esfuerzo en la solución de problemas dinamizando la interacción y discusión con sus pares y sus profesores.
3. Los estudiantes, en su mayoría, logran superar las dificultades de los retos y, fruto de la interacción persistente con los diferentes modelos, encuentran soluciones y buscan perfeccionarlas apropiándose en ese proceso de la teoría necesaria.
4. El material didáctico desarrollado ha evolucionado luego de múltiples iteraciones de uso, con los aportes de todos los participantes de la experiencia, logrando versiones cada vez más pertinentes y prácticas para el propósito de aprendizaje.
5. La interacción con los modelos de experimentación ha permitido encontrar invariantes novedosos que trivializan algunos de los retos planteados en los escenarios; esto ha concientizado a los estudiantes de la importancia real de este concepto antes tan abstracto.
6. La creación de escenarios para el área de programación bioinspirada les permite a los estudiantes aprender desde sus primeros semestres temas novedosos y útiles que actualmente se tratan en cursos especializados de posgrado.

Referencias

- Ausubel, D. P., Novak, J. D., & Hanesian, H. (1976). *Psicología educativa: un punto de vista cognoscitivo* (Vol. 3). México: Trillas.
- Albornoz, J. Chaparro, R. (2002). Escenarios de Aprendizaje. Programa de Investigación. Escuela Colombiana de Ingeniería Julio Garavito. Decanatura de Ingeniería de Sistemas.
- Albornoz, J. Chaparro, R. (2005). "The Learning of Fundamental Concepts and Problem Solving Strategies in Informatics", Proceedings of the 6th International Conference on Higher Education.
- Escuela Colombiana de Ingeniería (2010). Proyecto Educativo Institucional. Bogotá D.C.
- Casallas, C. Vargas, S. (2009). Bioescenarios IV Inteligencia de enjambres. Trabajo de grado. Escuela Colombiana de Ingeniería Julio Garavito. Decanatura de Ingeniería de Sistemas.
- Díaz, M. Díaz, F., & Pulido, G. (2011). Bioescenarios VI. Bee colony. In *Computing Congress (CCC), 2011 6th Colombian* (pp. 1-6). IEEE.
- Hodges, A. (1995). Alan Turing and the Turing machine. In *The Universal Turing Machine A Half-Century Survey* (pp. 3-14). Springer Vienna.

- Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. University of Chicago press.
- Radford, L. (2013). Three key concepts of the theory of objectification: knowledge, knowing and learning. *Journal of Research in Mathematics Education*, 2 (1), 7-44.
- Piaget, J. (1972) *Psicología y pedagogía*. Ariel. 1972.
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, (1), 33-57. *intelligence approach*. Springer Verlag.
- Vygotsky, L. (1995). *Pensamiento y lenguaje*. A. Kozulin (Ed.). Barcelona: Paidós.

Sobre los autores

- Albornoz Bueno, Juan. Ingeniero de sistemas y computación. D.E.A. Informática. Especialista en Construcción de Software de la Universidad de los Andes. Profesor de cátedra de la Escuela Colombiana de Ingeniería Julio Garavito. juan.albornoz@escuelaing.edu.co
- Chaparro Aguilar, Raúl Alfredo. Magíster en Matemáticas y en Ingeniería de Sistemas de la Universidad Nacional. Profesor de la Escuela Colombiana de Ingeniería Julio Garavito. raul.chaparro@escuelaing.edu.co
- Díaz Roza, María Irma. Ingeniera de sistemas y computación de la Universidad de los Andes, máster en Ingeniería de Software de la Politécnica de Madrid. Profesora de la Escuela Colombiana de Ingeniería Julio Garavito. maria.diaz@escuelaing.edu.co

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería y de la International Federation of Engineering Education Societies

Copyright © 2013 Asociación Colombiana de Facultades de Ingeniería (ACOFI), International Federation of Engineering Education Societies (IFEES)