



MODELO DE ENSEÑANZA Y SU RELACIÓN CON LOS PROCESOS METACOGNITIVOS EN PROGRAMACIÓN DE SISTEMAS¹

TEACHING APPROACH AND ITS RELATIONSHIP WITH META-COGNITIVE PROCESSES IN SOFTWARE PROGRAMMING

Cristina Romero Chaves y María Mercedes Rosero Sosa
Institución Universitaria CESMAG, Pasto (Colombia).

Resumen

Este artículo pretende dar cuenta del modelo de enseñanza aplicado por los docentes en las asignaturas de fundamentos de programación en Ingeniería de Sistemas y la relación que se establece con la aplicabilidad de procesos metacognitivos en los que se analizan categorías como la planeación, los objetivos, el modelo didáctico y la evaluación. La investigación utilizó el método de estudio de caso, desde la perspectiva de la acción situada, que se desarrolló en cuatro instituciones de educación superior de la ciudad de Pasto. Se concluye que el modelo de enseñanza aplicado se inscribe en el paradigma constructivista, con un modelo didáctico enfocado al aprendizaje basado en problemas, que no evidencia la intervención en procesos metacognitivos como la planeación, el control y la evaluación, sólo se orienta a desarrollar habilidades o conocimientos propios de la disciplina.

Palabras claves: aprendizaje basado en problemas; metacognición; modelos de enseñanza; modelos mentales; fundamentos de programación

Abstract

This article aims to explain the teaching approach implemented by teachers in the fundamentals of software programming subjects in computers engineering and the established relationship with the applicability of meta-cognitive processes, where categories such as planning, objectives, teaching approach and evaluation are analyzed. The research design used for this inquiry was the case study method, from the situated action perspective which was developed in four higher education institutions in San Juan de

¹ Artículo derivado de la investigación “Procesos metacognitivos basados en la teoría de la actividad del docente de Ingeniería de Sistemas”, del grupo de investigación Modelos Pedagógicos (Anselmo Caradonna). Investigación financiada y avalada por la Institución Universitaria Cesmág.

Pasto. The conclusion is that the teaching approach used in this study follows the constructivist paradigm, with a didactic problem-learning model that does not indicate the intervention in meta-cognitive processes such as planning, monitoring and evaluation, it only targets on developing proper skills or knowledge for this discipline.

Keywords: problem-learning model; meta-cognition; teaching model; mental models; fundamental programming

Introducción

Las diversas teorías sobre los modelos de enseñanza van desde los tradicionales y los constructivistas hasta los investigativos (Joyce y Weil, 1985; Shulman, 1986; Sacristán, 1994; García y Angulo, 2003; Rodríguez, 1999; Chevillard, 1991; Porlan, 1993, 1995, entre otros). Sin embargo, es importante aclarar que en las prácticas de enseñanza² universitaria no se determina un modelo único o ideal que copie de manera fotográfica sus postulados, por cuanto no se puede desconocer la realidad que acontece en el aula, el contexto, la interacción entre docente y estudiantes. Es decir, en el acto pedagógico se presentan características disímiles que de algún modo hacen prever más bien la predominancia de un modelo sobre otro con la utilización de estrategias didácticas específicas.

Es así como, a través de la historia, en la enseñanza de la programación han cohabitado diferentes enfoques y tendencias (Ferreira y Rojo, 2006). Actualmente se puede mencionar que no hay un consenso de métodos de enseñanza por su diversidad de orientación e igualmente por los enfoques didácticos aplicados.

En este sentido, Kaasbøll (1999) plantea tres modelos didácticos que han imperado en la enseñanza de la programación: el de la escalera semiótica, que hace referencia al uso de lenguajes de programación y búsqueda de instrucciones, que identifica una progresión de secuencias sintácticas, luego semánticas y finalmente, la pragmática del lenguaje como principal herramienta; el de objetivos de taxonomía cognitiva, encaminado al uso de secuencias a través de instrucciones, estrategia que guarda similitud con la taxonomía por objetivos de Bloom; y el de resolución de problemas; considerado más como un modelo de aprendizaje que una estrategia de enseñanza.

Sin embargo, en los modelos expuestos coexisten problemáticas referidas a dificultades en el aprendizaje de la programación (McCracken, Almstrum, Díaz et al, 2001; Milne y Rowe, 2002; Robins, Rountree y Rountree, 2003; Azpilicueta y Ledesma, 2004; Villalobos, Casallas y Marcos, 2005; Checa, 2011; Ala-Mutka, s.f.) y a dificultades en la enseñanza, por lo que algunos se interrogan acerca de eficiencia y eficacia de los lenguajes propuestos y los entornos de aprendizaje (Pane y Myers, 1996), la comprensión (Whalley, Lister, Thompson et al, 2006), la resolución de problemas (Du Boulay, 1986; Lister, Adams, Fitzgerald et al, 2004), el manejo de lenguajes y de idiomas (Leavens, 2008), entre otros.

Por otra parte, la metacognición favorece los procesos de enseñanza y aprendizaje. Los modelos de enseñanza pueden favorecer la metacognición, entendida como el conocimiento sobre el conocimiento. Al respecto, hay numerosas investigaciones que la sustentan (Baker y Brown, 1984; Flavel, 1976, 1979; Garrison, 1997; Hacker, Dunlosky y Graesser, 1998; Paris y Winogrend, 1990; Schraw, 1998; Kuhn, 2000; Schraw y Dennison, 1994, citado por Balcikanl, 2011) y menciona que es un conocimiento de las cogniciones mediante distintas operaciones mentales relacionadas con la atención, la percepción, la comunicación, la memorización, la lectura y la escritura y la capacidad de regular de forma deliberada su actividad mental (Gombert, 1990; Romainville, 2000).

Ahora bien, en las ciencias de la programación, los resultados de las investigaciones de Volet & Lund³ (citado en Miles, Blum, Staats y Dean, 2003) señalan que el aprendizaje de los estudiantes no debe subsumirse al nivel sintáctico del conocimiento, sino pasar al nivel de conocimiento conceptual, unido al desarrollo de habilidades cognitivas básicas y

² Véase Romero (2010) Constitución de sujeto profesor a través de las prácticas de enseñanza. Revista Universidad Católica del Norte, 31, p.148.

³ Véase Volet, S. & Lund, C.P. (1994). Metacognitive instruction in introductory computer programming: a better explanatory construct for performance than traditional factors. Journal of Educational Computing Research, 10 (4). pp. 297-329.

al uso de habilidades metacognitivas dirigidas a la resolución de nuevos problemas de programación, programas de prueba y depuración de errores lógicos (Miles et al., 2003). También indican la importancia de incrementar en los estudiantes una conciencia de sus propios modelos mentales, una descomposición de la estructura jerárquica y relacional entre los objetos y una evaluación del progreso de cada estudiante.

Metodología

La investigación se inscribe en el paradigma naturalista con un enfoque histórico-hermenéutico. Se utilizó como método el estudio de caso y en el campo educativo se atendió a aportes de Durant (citado en Pradas, 2010) en cuanto a la acción situada, la que posibilita articular la acción y la situación mutuamente influenciada y corresponsable de un proceso continuo que permite entender la acción del profesor en coherencia con la situación.

Se tomaron como unidad de análisis diez docentes que orientaban la asignatura de Fundamentos de Programación de Ingeniería de Sistemas en la Universidad de Nariño, la Universidad Mariana, la Fundación San Martín y la Institución Universitaria Cesmag. La unidad de muestreo correspondió a cuatro docentes con un tipo de muestra por conveniencia que compartió criterios como igual semestre académico y 80 % de contenidos similares.

Técnicas e instrumentos de recolección de la información

Se aplicó una entrevista no estructurada a los cuatro docentes que están a cargo de la asignatura de Fundamentos de Programación. Se realizó la observación directa a los docentes en mención y a los estudiantes durante la clase, por un periodo de cuatro meses. Los temas contemplados en la guía de cuestionario y el diario de campo se relacionaron con los siguientes aspectos: modelo didáctico empleado y criterios para aplicarlo, planificación que el docente desarrolla en cuanto a procesos metacognitivos, análisis de actividades y operaciones requeridas para el desarrollo de los temas previstos para las clases, acciones utilizadas para identificar y corregir los errores de los estudiantes, manera como se desarrollan los procesos de pensamiento requeridos con los estudiantes (análisis, síntesis, memoria, atención,

etc.), los criterios para elaborar ejercicios en clase y talleres, estrategias didácticas empleadas que articulan los contenidos, recursos que utiliza el docente (libros, lecturas, guías, videos, *software*, etc.), evaluación de aprendizajes y procesos de retroalimentación.

Aspectos metodológicos de análisis e interpretación de resultados

Para el análisis e interpretación de resultados se utilizó la categorización y la triangulación (Danzini, 2009; Thurmod, 2001) como procedimientos metodológicos que validaron la investigación desarrollada. Se realizó una categorización deductiva e inductiva, lo que permitió profundizar el análisis del problema de investigación. Para analizar el modelo de enseñanza se utilizó el procedimiento de categorización deductiva atendiendo a las subcategorías de planificación, objetivos, modelo didáctico y evaluación.

Resultados y discusión

Para desplegar el análisis categorial se decidió interpretar de manera individual las subcategorías planificación y evaluación. En el caso de objetivos y modelo didáctico se procedió a vincularlas, dado que la información obtenida tiene una relación muy estrecha.

- **Planificación.** De acuerdo con la manera en que se planifica la asignatura o espacio académico, los docentes indican que se realiza mediante un programa previamente estructurado denominado guía académica, programa analítico o programa guía, que en esencia retoma aspectos relacionados con un plan de organización, ejecución y evaluación de la asignatura para aplicarlo durante un semestre académico.

El contenido del plan presenta una estructura general y otra particular. En la estructura general, la planificación es una actividad inherente, anticipada al proceso de enseñanza desarrollada en cada semestre, cuya connotación es propia de los docentes y de las características académico-administrativas de los programas, sin olvidar que algunos programas son analizados previamente por el comité curricular, éstos corresponden a programas diseñados anteriormente por otros docentes los que se toman como guía para armar lo propuesto por el profesor y en otro caso, responde

a una propuesta investigativa que han tenido éxito a través de la implementación de un modelo de enseñanza de la programación, como sucede en el caso del proyecto Cupi2⁴, aplicado por la Universidad de los Andes (Colombia).

El dispositivo que enmarca un tipo de enseñanza y aprendizaje deseable en los estudiantes, lo constituye la planificación que direcciona unas finalidades explícitas acordes al planteamiento de objetivos, el perfil ocupacional y la realidad contextual, cuya dinámica debe responder a resolver interrogantes acerca del qué, cómo, cuándo y para qué se enseña y se dispone a través de unos procesos organizados secuencialmente. Si se analiza la primera estructura, se infiere que son constructos básicos y abstractos a través de los cuales se planifica la asignatura.

De otro lado, el tipo de planificación de la *estructura específica* no es aplicado por todos los docentes, éste se lleva a efecto mediante la organización de un plan de actividades dispuestas en un cronograma. Vale la pena mencionar que las estrategias, condiciones y fechas de evaluación se aplican de manera conjunta entre el docente y los estudiantes. Al revisar los programas denominados ficha de desarrollo temático y planeación de actividades se observa que las actividades a desarrollar tienen una relación conexas entre tema, propósito, recursos y evaluación, las que son tratadas de forma pormenorizada.

El modelo de aprendizaje basado en problemas (ABP) se evidencia en algunos apartes de las acciones propuestas por los docentes; no obstante, no se encuentra explícito un proceso de planificación relacionado con estrategias metacognitivas, sólo se orientan a desarrollar habilidades o conocimientos propios de la disciplina.

Por otra parte, en uno de los casos investigados se encontró que la aplicación del proyecto Cupi2 presenta cuatro clases de programación supeditadas a cinco niveles acumulables; en cada uno de ellos el estudiante debe superar los objetivos

propuestos para avanzar al siguiente nivel. Se resalta que los programas (clase, laboratorio, ejercicios y evaluación) tienen una secuencia de enseñanza vinculante a objetivos, conceptos, tareas y actividades de regulación que se analiza en el modelo didáctico.

Los mecanismos de control de la planificación de la asignatura, que tienen que ver con las estrategias de regulación de la enseñanza utilizada por los docentes, se entienden más bien como un proceso que parte de la experiencia, del manejo de la disciplina, el reporte del tema y la estrategia de evaluación registrada semanalmente en una ficha de control académico-administrativo y el seguimiento estricto de una metodología de planificación, para este último caso propuesto por el proyecto Cupi2.

Lo anterior indica que no hay un mecanismo introspectivo de evaluación acerca de la planificación por parte de los docentes, entendido como un “proceso de autorregulación” (Sanmarti citado por Angulo, 2002, p. 122) que lleva a unos cambios en la toma de decisiones y control de la planificación, lo que le permite al docente analizar, seleccionar, personalizar o proponer nuevas estrategias, plantear recursos o ratificar procedimientos.

- **Entre el modelo didáctico y los objetivos.** Es importante mencionar que el modelo didáctico tiene un estrecho acercamiento al modelo constructivista, en el que opera específicamente el ABP. Los docentes centran su proceso de enseñanza en la solución de problemas de la vida cotidiana a los que se enfrentará el futuro ingeniero de sistemas en materia de programación; en estas estrategias proponen ejercicios que se han diseñado y seleccionado previamente por parte del docente.

Al respecto, en las clases se observa que el profesor les solicita a los estudiantes que se remitan a los ejercicios dispuestos en el texto o guía, para que los resuelvan en el momento y posteriormente a cada grupo de estudiantes les asigna un ejercicio determinado para solucionarlo de manera individual.

Al confrontar los procesos metodológicos utilizados por el docente para abordar los conceptos objeto de enseñanza, mediante la resolución de problemas, se identifica que la secuencia didáctica gira en torno a:

⁴ Véase Villalobos, J. (2009). Proyecto Cupi2, una solución integral al problema de enseñar y aprender a programar. Bogotá: Universidad de los Andes.

- **Explicación por parte del profesor mediante la solución de problemas demostrativos:** a pesar de que se indican el tema por aprender y en la resolución del problema se especifican los datos de entrada y salida y el proceso, no se alcanza a evidenciar el mecanismo cognitivo que usa el docente, entendido como el modelo o la representación mental para ser comunicado verbalmente a partir del conocimiento procedimental. Este punto coincide con lo expuesto en la problemática de enseñanza de la programación expuesta por Villalobos (2009, p, 10): “Se da la sensación de que el ‘cómo hacer las cosas’ no es enseñable, y que es algo que depende de la inspiración y del ingenio del programador”; además, el autor manifiesta que en las evaluaciones esta dinámica genera ansiedad, debido a que el estudiante no sabe si el modelo que tiene en su estructura cognitiva es suficiente para resolver el problema, el que se entiende como algo que llega mágicamente a su cabeza y no como un proceso consciente de autorregulación.
- **Ejercicios dirigidos a resolver problemas:** el diseño de un bagaje de ejercicios propuestos en la fase de planificación por el profesor y la resolución de los mismos por parte de los estudiantes, corresponde a un proceso metodológico característico de la enseñanza de la programación. El objeto de este recurso didáctico se orienta a la aplicación de conceptos tratados en la anterior fase y al desarrollo de las habilidades en lectura, escritura, el manejo de vocabulario, interpretación, comprensión, etc. Este proceso se lleva a cabo en dos momentos; uno en el aula, de manera individual (en pocos casos se trabaja en grupos) y con la asesoría del docente, y el otro son ejercicios que se dejan para resolverlos en el tiempo independiente. Cabe restar que la resolución de problemas no limita la aplicación de otras estrategias de enseñanza y aprendizaje que pueden acompañar y apoyar los procesos metacognitivos.

Una de las tareas que precisa la aplicación del proceso metacognitivo estaría representada en la solución del problema sin el computador; este espacio constituye el punto crucial que desde el modelo de enseñanza se debe intervenir de una manera más específica, por cuanto es aquí donde se construyen los modelos mentales de los estudiantes: procesos que utilizan, explicitación del modelo, desarrollo de habilidades de reflexión, dificultades encontradas, aspectos no resueltos que llevan a la internalización y externalización del conocimiento.

La explicitación del modelo mental del profesor y de los estudiantes es un paso muy importante en el proceso de enseñanza de la programación, ya que es un mecanismo que les ayudaría a los estudiantes a comprender con mayor profundidad los procesos cognitivos que se requieren en la solución de problemas. No basta con generar una dinámica propia secuencial y detallada por fases, sin incluir estrategias metacognitivas que favorecería no sólo la enseñanza sino el aprendizaje.

La explicación que hace uno de los docentes a los estudiantes desde la metacognición es muy válida y apropiada, por cuanto hace explícita la utilización de los modelos mentales (Solaz-Portóles & López, 2007) como estrategia que potencia el razonamiento y la toma de decisiones, y a su vez activa procesos de pensamiento y posibilita la autorregulación de los aprendizajes en los estudiantes; sin embargo, el docente no es consciente de la utilización de procesos metacognitivos. En otra de las observaciones se detecta que el profesor no les explica a los estudiantes el modelo mental que utilizó en la resolución de un problema, por lo cual éstos no podrán utilizar esos mecanismos en forma consciente en la resolución de problemas.

De igual manera, se considera pertinente revisar la situación problemática que se genera en el proceso enseñanza. Ello conduce a que el docente revise los presupuestos pedagógicos investigativos y prácticos vinculados con el modelo didáctico empleado por el profesor mediante procesos intelectuales y lingüísticos. Si bien es cierto que los cuatro docentes aplican actividades que favorecen la solución de problemas, surgen los siguientes interrogantes que el docente debe revisar: ¿qué modelo didáctico emplea para la enseñanza de la programación?, ¿qué proceso metodológico se está llevando a cabo en el planteamiento, desarrollo y solución de problemas?, ¿qué instrumentos de pensamiento relacionados con nociones, proposiciones y conceptos se están planteando con los ejercicios de solución de problemas? y ¿qué modelos

mentales están desarrollando los estudiantes con los ejercicios planteados?

Se evidencia que el modelo didáctico en su parte conceptual es entendido como una estrategia de aprendizaje; precisamente el ABP determina un punto clave para el análisis de condiciones que demarcan un proceso de enseñanza, el cual va desde la explicación del profesor hasta la aplicación de conceptos, técnicas y manejo de un lenguaje de programación. Por lo tanto, la resolución de problemas es una estrategia que implica una tarea cognitiva y sociocognitiva y, según Montealegre (2007, p. 24), se requiere que una persona posea “conocimiento lingüístico, semántico, esquemático (conocimiento de los tipos de problemas, conocimiento operativo (como llevar a cabo la secuencia de operaciones o conocer el algoritmo exacto para llevar a cabo la tarea) y estratégico (técnicas para resolver un problema”.

Es decir, no basta que el estudiante conozca y haga uso del concepto sino que el docente esté consciente de comprender, planear y ejecutar mecanismos que medien el desarrollo de la actividad cognitiva en la resolución de un problema; sin embargo, queda la inquietud acerca de cómo se hace uso del proceso didáctico para que los estudiantes adquieran conocimientos. A pesar de que los registros de observaciones de clases demuestran consonancia entre la explicación del concepto por abordar por parte del docente y posteriormente la aplicación de ejercicios a cargo de los estudiantes, el profesor refuerza el aprendizaje orientando a cada estudiante o grupos de estudiantes para resolver dudas o corregir errores en el momento, pero en muchos casos se queda solo en este paso, sin profundizar en otra clase de conocimientos como los planteados por Montealegre.

Al continuar las evidencias que caractericen el o los modelos de enseñanza aplicados se recurre a otro registro de entrevista, un docente comenta cómo fue aplicando el modelo didáctico en su quehacer pedagógico y referencia el paso de un modelo de enseñanza tradicional a un constructivista que ha demandado una variedad de experiencias, de formación, de reflexión, entre otros, fruto de una acción consciente del profesor en la búsqueda de opciones que mejoren la enseñanza. Para tal

fin enuncia la *negociación*, entendida como un dispositivo detonante en la construcción mental que hace el estudiante.

Desde el constructivismo social se considera que los ambientes de aprendizaje relacionados con el trabajo colaborativo favorecen la interrelación entre personas y entornos que impliquen crear ambientes propicios de negociación social y el impulso de la capacidad reflexiva individual (Gruba y Lynch, 1997), de este modo el concepto de negociación utilizado hace referencia a la relación que se establece entre el docente y el estudiante; o sea que el aprendizaje con la ayuda de otros o lo que Vygotski (1989) denominó *zona de desarrollo próximo* (ZDP), lugar en el que emerge un doble movimiento, por un lado la construcción de modelos mentales que el estudiante ha creado a partir de la utilización de dispositivos (lingüísticos, tecnológicos, psicológicos, etc.) en la solución de un problema y, por el otro, la negociación a la que se llega con el docente.

Cabe precisar que desde la psicología cognitiva y principalmente desde la enseñanza de ciencias se han realizado estudios acerca de los modelos mentales, lo cuales se entienden como representaciones analógicas que subyacen a una realidad a partir de una situación determinada (Johnson-Laird, 1987). Desde otro referente se define como el acoplamiento de construcciones humanas que se gesta en la mente de todo sujeto de forma individual o personal (Moreira, 1999).

Por lo tanto, se puede señalar que su aplicabilidad tiene fundamento desde dos vertientes: la primera, favorecería el aprendizaje de los estudiantes, de forma consciente evocarían procesos utilizados para solucionar los problemas, y la segunda vertiente, desde la enseñanza, le brindaría al docente la oportunidad no sólo de conocer criterios y formas de modelos mentales sino intervenir en la planeación, ejecución y control de las actividades de enseñanza dirigidos a fortalecer, ampliar o reestructurar las representaciones mentales de los estudiantes.

De acuerdo con lo expuesto en las entrevistas, es necesario profundizar en el modelo de enseñanza constructivista enunciado por los docentes e ir

articulando con las observaciones de clases en función de unas subpreguntas que potencian el primer interrogante y del cual emergen subcategorías deductivas como, por ejemplo, el ABP y los modelos mentales. Posteriormente se continuó con la siguiente pregunta ¿cuáles son los criterios que se establecen para el desarrollo de los ejercicios en clase, los talleres y trabajos?

Vale la pena mencionar que, de acuerdo con este interrogante, cada docente tomó un criterio distinto. Por lo tanto, las investigadoras hacen referencia a todos ellos, enunciando que los tres primeros se tratarán en este aparte, los cuales hacen referencia a los modelos mentales y la motivación. Finalmente, el tercer criterio se aborda en la siguiente subcategoría que corresponde a evaluación.

Los registros de observaciones de clases se conectan de cierta manera con el trabajo y orientación individual, y se convierten en una constante de la metodología aplicada para los ejercicios, talleres o trabajos, aunque sólo uno de los docentes mencionó este criterio. En la práctica se presenta en todos los grupos de estudio, aclarando que no es la única. De ahí que es válido plantear que para aprender a programar se requiere el desarrollo de la actividad cognitiva expuesta a través de representaciones mentales que le permitan al docente evaluar el análisis que ha realizado el estudiante. Esta actividad demanda tanto el procesamiento de la información como la forma en que el estudiante soluciona el problema.

No obstante, para uno de los profesores, el trabajo colaborativo se ve desplazado de cierta manera como estrategia didáctica, si bien le posibilita al estudiante, con la ayuda de otros pares, mejorar su proceso cognitivo a través del diálogo y la interacción, y no se puede supeditar únicamente a que el estudiante encuentre la solución. Precisamente éstas son herramientas potenciadoras para resolver un problema en el proceso y no al final. Además, con estas actividades los estudiantes pueden hacerse responsables de su aprendizaje y ayudar a los demás a movilizar sus representaciones mentales.

Desde la perspectiva socioconstructivista de la enseñanza y el aprendizaje, los avances del grupo

de investigación Grintie, liderado por César Coll (1984), brindan posibilidades de análisis en cuanto a la interactividad de docentes y estudiantes sobre una base epistemológica y pedagógica de la organización social de las actividades de aprendizaje, desde las cuales se indican posturas teóricas e investigativas en las que subyacen tres formas de organización social: cooperativa, competitiva e individualista. La caracterización de las anteriores formas de organización social se aplica en función de la interdependencia entre los estudiantes y la tarea por efectuar o el objetivo por alcanzar y el aprendizaje operante como lo indican Lewin, Deutsch y Johnson (citados por Coll, 1984, p. 120-121).

Al comparar las tres formas de organización social de aprendizaje dispuestas por Coll (1984), con el criterio expuesto por el docente se puede inferir que existe una mayor relación en la organización social de las actividades de aprendizaje con la individualización, aspecto que se corrobora cuando el docente menciona que el estudiante primero debe desarrollar individualmente su capacidad de análisis en el ejercicio y posteriormente realizar el estudio con otros pares o compañeros, aunque vale anotar que el desplazamiento hacia la organización cooperativa se vislumbra desde una vía que hace mención a la relación establecida entre docente y estudiante, a partir de un vínculo interactivo y dialógico que se suscita cuando el docente le solicita al estudiante la evocación y demostración de su representación mental.

Lo anterior lleva a considerar la necesidad de propiciar la organización cooperativa sin demeritar que en ciertos momentos se requiere de la individualista, por lo que vale la pena mencionar la tarea que hace el docente en las clases de programación a partir de las actividades que se dejan en los talleres o trabajos: identifica errores, corrige, le explica no sólo al estudiante que los comete sino a los demás compañeros, señala las dificultades o los aciertos, lo cual se utiliza desde la perspectiva constructivista como una posibilidad de aprendizaje, como se registró en las observaciones de las clases.

En la resolución de problemas desde la perspectiva socioconstructivista es importante que el docente no sólo mencione y explique el error, sino que

sean los mismos estudiantes los que identifiquen las dificultades, analicen las operaciones y regulen sus propios procesos. En las clases de programación, después de que el docente nota el error y les pregunta a los estudiantes su opinión, no se maximiza como una actividad pedagógica que aplique estrategias metacognitivas y lleve a reconocer el error.

Parece oportuno mencionar que, de acuerdo con las observaciones, los docentes retoman y aprovechan el error como un aspecto dirigido a lograr la respuesta correcta en la que se hace uso de la demostración y explicación de forma verbal y gráfica con respecto a la solución del problema, aplicando un modelo de reglas propias del docente. No se hace énfasis en una mayor exploración de otras formas de solución que contribuyan a la construcción del conocimiento, en el error opera el diálogo triádico (Lemke, 1997) distinguido por el manejo de la pregunta por parte del profesor, la respuesta de los estudiantes y la retroalimentación de los presupuestos considerados y dirigidos únicamente desde la visión del docente.

En consecuencia, se puede mencionar que existen obstáculos didácticos en la resolución de problemas (D'amore, Hammar, & Tsapalis, citado en Gil, 2001, p. 86), debido a la persistencia de una serie de dificultades o hábitos. Tales errores se manifiestan cuando:

ejecutan operaciones rápidamente con errores, son inexactos en su interpretación, concluyen el problema sin razonar lo suficiente, no vinculan los conocimientos de la materia con el problema, no aclaran sus razonamientos para identificar las partes más difíciles, no relacionan lo escrito en el texto con la realidad y pasan inadvertidas palabras y frases que no entienden y que son claves.

En programación se podría decir que dichos errores se centran en la interpretación, el manejo de conceptos, la aplicación y relación de conocimientos disciplinares al problema, el manejo de procesos de razonamiento, planificación y análisis de las operaciones. Estos planteamientos, contextualizados desde la enseñanza y aprendizaje de la programación, se relacionan con las dificultades

que deben asumir tanto el docente como los estudiantes en los ejercicios propuestos.

Los elementos que se cruzan entre el saber y las habilidades requeridas en fundamentos hacen alusión a falencias en la manera de enseñar programación. Según Villalobos (2009), el docente plantea modelos o patrones generalizables por ser aplicados en los problemas con la pretensión de que el estudiante asuma el manejo de reglas y técnicas de resolución a los ejercicios propuestos por el docente, aplicando un modelo de imitación. El profesor, al disponer de una serie de técnicas para resolver variados tipos de problemas, permite que los estudiantes no se enfrenten a un problema cuyo algoritmo de resolución desconozcan; no obstante, la dificultad radica cuando emergen nuevos tipos de problemas que no manejan el mismo comportamiento de resolución o de técnica. Así los estudiantes no podrán transferir la técnica aprendida, como lo manifiesta Oñorbe (citado por Varela, 2002). Por otra parte, en los estudiantes se observa que cuando asumen un problema no detallan y hacen un análisis de cada tarea, simplemente se trasladan a la resolución de los ejercicios que parten de la explicación del profesor, sin hacer uso del conocimiento procedimental.

En la metacognición, el monitoreo de la actividad se considera un punto necesario a trabajar, que debe resaltarse en las observaciones realizadas a los docentes, ya que permite la retroalimentación, revisión de dificultades o corrección de errores. Precisamente este monitoreo se entiende como la toma de conciencia que tiene el sujeto sobre un determinado conocimiento; sin embargo, la tarea del docente (Otero, Campanario y Hopkins, 1992) consiste en propiciar que los estudiantes detecten el error como una tarea consciente y propia en la que a la vez expresen sus dificultades y traten de dar solución por sí mismos a los problemas planteados.

Lo anterior llevaría a indicar que existe un panorama claro acerca de cuáles son las competencias por desarrollar en este espacio académico; sin embargo, la función del profesor no puede reducirse a desarrollar modelos constructivistas que favorezca solamente procesos cognitivos, sino a incursionar

de forma amplia en estrategias metacognitivas para que los estudiantes autorregulen su aprendizaje.

Otro criterio propuesto por un docente menciona la motivación permanente y el desarrollo de su capacidad mental mediante el análisis y cómo desde los primeros semestres se estructuran las bases de la programación, dirigida a su formación ocupacional.

Evaluación

La retroalimentación es una actividad que los docentes realizan continuamente y está vinculada a la evaluación, que no sólo se entiende como el proceso que resulta de una acción cuantitativa para obtener notas parciales o finales, sino que es un proceso permanente, de cada clase, como lo señalan los docentes. Es importante identificar que la retroalimentación es un aspecto inherente a la evaluación por cuanto permite conocer a los estudiantes su desempeño, sus procesos de aprendizaje y la forma como se pueden potencializar o favorecer, lo que la convierte en una herramienta vital para el profesor.

Otro de los docentes precisa que la actividad de enseñar la regula en dos sentidos: una hacia sí mismo, en la que el control lo asume como una estrategia didáctica de regulación de la enseñanza, en la cual el mecanismo de evaluación difiere del proceso de valoración que hace en los laboratorios con el proyecto Cupi2 para asumir uno propio; y otra hacia los estudiantes. Éste es un aspecto que en las observaciones de clase se orienta a identificar los errores y la forma como se debe corregir, pero por el número de estudiantes no se profundiza en tales estrategias. Para Sanmarti y Jorba (2008) la regulación del aprendizaje se conecta con la evaluación y las estrategias didácticas que se emplean.

Por otro lado, la evaluación responde a procesos académico-administrativos que buscan realizar un acuerdo entre las partes en cuanto a fechas y eventos. Como bien lo denomina el docente, se establecen unas políticas claras desde el principio del semestre, realidad que forma parte de las políticas y condiciones que subyacen a la dinámica interna de las instituciones de educación superior a las que el docente debe responder. La función pedagógica de la evaluación se alcanza a observar en las clases a través del desarrollo

de los ejercicios o talleres y en los mismos eventos evaluativos.

De acuerdo con lo mencionado por uno de los profesores, se encuentra una relación entre el concepto por aprender y la evaluación, en tanto esta última se aprovecha como un espacio didáctico y a la vez de verificación de los aprendizajes en los que se está fallando; no obstante, la evaluación no se puede comprender únicamente como calificación, pues en este proceso se puede buscar otras opciones que favorezcan el desarrollo del análisis y la síntesis y básicamente de la construcción y afianzamiento de los modelos mentales.

Si bien es cierto que la evaluación es una expresión que implica valorar el desempeño del estudiante, orientado a juzgar el proceso de aprendizaje, la tarea del docente debe remitirse a la adecuación de la enseñanza y la medida en que los objetivos de enseñanza se han cumplido. En este sentido, los estudiantes deben conocer los objetivos de aprendizaje de cada unidad didáctica, no como simple información sino como una reflexión de capacidades y tareas necesarias. Ahora bien, si existen dificultades para comprender un concepto, el acto pedagógico se debe aprovechar para buscar las herramientas cognitivas necesarias y los instrumentos que la favorezcan; es menester del profesor regular esta actividad y después analizar la eficacia de evaluar en los puntos en los que falla el estudiante.

Vale precisar que las herramientas cognitivas, desde la perspectiva de Jonassen et al. (citado en Martínez, Montero, Pedroza y Martín, s.f) hacen mención a las tecnologías de carácter tangible e intangible, que favorecen la cognición del individuo durante el pensamiento, la resolución de problemas y el aprendizaje, lo cual significa que con las actividades de refuerzo se pueden generar las denominadas herramientas cognitivas como parte de las prácticas de enseñanza que usa el profesor de programación.

Para ello, resulta imprescindible que el docente haga una representación de los objetivos de su quehacer, se apropie y aplique los procesos metacognitivos y determine la efectividad de sus acciones al resolver una tarea, lo cual le permite reconocer, evaluar y tomar decisiones respecto de sus propias ideas sobre qué y cómo enseñar, mientras diseña su acción didáctica.

Conclusiones

El modelo de enseñanza aplicado por los cuatro programas de ingeniería en la enseñanza de los fundamentos de la programación se inscribe en el paradigma constructivista, con predominancia del modelo didáctico del aprendizaje basado en problemas.

Los resultados muestran que los profesores no generan un proceso de reflexión en la resolución de problemas, que les permita tomar en consideración las condiciones de la situación de enseñanza y anticiparse a las respuestas de sus estudiantes.

No se evidencian procesos de autorregulación aplicados al modelo de enseñanza que favorezcan la intervención en procesos metacognitivos como la planeación, el control y la evaluación; más bien se encuentra que la experiencia del profesor o los lineamientos de políticas académico administrativas son orientadores fundamentales de dicho proceso.

No hay mecanismos introspectivos de evaluación acerca de la planificación, entendidos como autorregulación. En sí, la toma de decisiones y control de la planificación que permitiría a los docentes analizar, seleccionar, personalizar o proponer nuevas estrategias, plantear recursos o ratificar procedimientos. La secuencia didáctica gira en torno a la explicación por parte del profesor mediante la solución de problemas demostrativos y no se alcanza a evidenciar el mecanismo

cognitivo del que hace uso el docente, entendido como el modelo o la representación mental para ser comunicado verbalmente a partir del conocimiento procedimental.

Lo anterior llevaría a plantear que existe un panorama claro acerca de cuáles son las competencias por desarrollar en este espacio académico. Sin embargo, la función del profesor no puede reducirse a desarrollar modelos constructivistas que favorezcan solamente procesos cognitivos; debe incursionar ampliamente en estrategias metacognitivas para que su labor y la de los estudiantes constituya su propia autorregulación.

La resolución de problemas, como metodología propia de la enseñanza de la programación, debe ser intervenida con procesos que modifique el manejo de los modelos mentales, el aprendizaje de lenguajes, la interpretación y la comprensión.

Las prácticas metacognitivas son esenciales para promover los cambios apropiados en el desarrollo del docente de ingeniería de sistemas, no sólo en lo relativo a sus ideas acerca de la enseñanza y el aprendizaje y los papeles de docente y estudiante, sino también acerca del contenido disciplinar, sus habilidades y los fundamentos epistemológicos básicos que debe dominar. A la vez, las estrategias metacognitivas se consideran adecuadas para desarrollar habilidades de anticipación y planificación de las acciones al resolver una tarea.

Referencias

- Ala-Mutka, K. (s.f). *Problems in learning and teaching programming - a literature study for developing visualizations in the Codewitz-Minerva project-Codewitz needs analysis*. Recuperado el 1.º de noviembre de 2013 de <http://www.ijiet.org/papers/247-T0030.pdf>.
- Angulo, F. (2002). *Aprender a enseñar ciencias: análisis de la aplicación de una propuesta basada en la autorregulación de los aprendizajes*. Tesis para la obtención del título de doctor en Matemáticas, Departament de Didàctica de Matemàtica i les Ciències de l'Educació de la Universitat Autònoma de Barcelona, España.
- Azpilcueta, J. & Ledesma, A. (2004). *Constructivismo en la educación de las ciencias de la computación. Una propuesta de enseñanza-aprendizaje en aula virtual*. Recuperado el 1.º de noviembre de 2013 de <http://www.ijiet.org/papers/247-T0030.pdf>.
- Balcikanlı, C. (2011). Metacognitive Awareness Inventory for Teachers (MAIT). *Electronic Journal of Research in Educational*, 9(3), 1309-1332.
- Checa Fernández, R. (2011). La innovación metodológica en la enseñanza de la programación. Una aproximación pedagógica al aprendizaje activo en la asignatura Fundamentos de Programación. *Revista Digital de la Facultad de Ingeniería de Sistemas- Interfaces*, (4), 67-87.
- Chevallard, Y. (1991). *Del sabio al saber enseñado*. Buenos Aires: Aique.
- Coll, C. (1984). Estructura grupal, interacción entre alumnos y aprendizaje escolar. *Infancia y Aprendizaje*, 27(28), 119-138.

- Danzini, N. (2009). *The research act: a theoretical introduction to sociological methods*. Chicago: library of Congress Cataloging-in. Publication Data.
- Du Boulay, B (1986). Some difficulties of learning to program. *Revista de Investigación Educativa Informática*, 2(1), 57-73.
- Ferreira, S. y G. Rojo. (2005). *Cambios metodológico didácticos y evaluación del impacto de los mismos en un curso introductorio a los conceptos de algorítmica y programación*. En Proceedings de las Primeras Jornadas de Educación en Informática y TIC en Argentina, Bahía Blanca, Argentina, pp.210-216.
- Ferreira, A. & Rojo, G. (2006). Enseñanza de la programación. TE&ET. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, 1(1). Recuperado el 18 de noviembre de 2012 de http://teyet-revista.info.unlp.edu.ar/files/Nol/09_Ensenanza_de_la_programacion.pdf.
- García, P. y Angulo, F. (2003). Un modelo didáctico para la formación inicial del profesorado de ciencias. *Revista Interuniversitaria de Formación de Profesorado*, 17(001), 37-49.
- Gil, R. (2001). *La actividad metacognitiva como desencadenante de procesos de autorregulación en las concepciones y prácticas de enseñanza de los profesores de ciencias experimentales*. Tesis para la obtención del título de doctor en Matemáticas, Departament de Didàctica de Matemàtica i les Ciències Experimentals de la Universitat Autònoma de Barcelona Barcelona, España.
- Gombert, J. (1990). *Le développement métalinguistique*, París: PUF.
- Gruba, P. y Lynch, B. (1997). Constructivist approaches to communication skills instruction. Consultado el 18 de noviembre de 2009 de <http://www.ascilite.org.au/conferences/perth97/papers/Gruba/Gruba.html>.
- Johnson-Laird, P. (1987). Modelos mentales en ciencia cognitiva. En Norman, D. *Perspectivas de la ciencia cognitiva. Cognición y desarrollo humano*, pp 179- 232. Barcelona: Paidós.
- Joyce, B.R. y Weil, M. (1985). *Modelos de enseñanza*. Madrid: Anaya.
- Kaasbøll, J. (1999). *Exploring didactic models for programming*. Recuperado el 11 de enero de 2011 de <http://citeseer.nj.nec.com/330187.htm>.
- Laxer, C., Thomas, L., Utting, I. y Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students, *SIGCSE Bulletin*, 33(4), pp. 125-180.
- Leavens, G. (2008). Use concurrent programming models to motivate teaching of programming languages. *ACM SIGPLAN Notices*, 43 (11), 2008, 93-98.
- Lekme, J. (1997). *Aprender a hablar ciencia. Lenguaje aprendizaje y valores*. Barcelona: Paidós.
- Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J. et al. (2004). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36(4), pp. 119-150.
- Miles, D. Blum, T., Staats, W.J. y Dean, D., (2003) Experiences with the metacognitive skills inventory. *Frontiers in Education, FIE 2003 33rd Annual* , (1), pp.T3B,8-13, (1), 5-8. doi: 10.1109/FIE.2003.1263324.
- Milne, I. y Rowe, G. (2002). Difficulties in learning and teaching programming. *Education and Information Technologies*, 7(1), pp. 55-66.
- Montealegre, R. (2005). La actividad humana en la psicología histórico-cultural. [Versión electrónica], *Avances en Psicología Latinoamericana*, (23), pp. 33-42.
- Otero, J., Campanario, J. & Hopkins, K. (1992). The relationship between academic achievement and metacognitive comprehension monitoring ability of spanish secondary school students. *Educational and Psychological Measurement*, (52), pp. 419-430.
- Pane, John y Myers, Brad, (1996). Usability Issues in the Design of Novice Programming Systems. Institute for Software Research de <http://repository.cmu.edu/isr/820>.
- Pérez, G. (1994). Enseñanza para la comprensión. En Sacristán, G. y Pérez, G. *Comprender y transformar la enseñanza* (3.ª ed.), pp. 78-107. Madrid: Morata.
- Porlan, R. (1995). Qué y cómo enseñar desde una perspectiva constructivista. En Mariño, G. (Comp.) *Constructivismo y didáctica*. Bogotá: Dimensión Educativa.
- Porlan, R. (1993). *Constructivismo y escuela. Hacia un modelo de enseñanza aprendizaje basado en la investigación*. Sevilla: Diada Editora.
- Pradas, R. (2010). El estudio del pensamiento del profesorado desde la perspectiva de la acción situada en la educación física escolar. *Cultura y Educación*, (22), 21-36.
- Moreira, M. (1997). Modelos mentales. *Investigações em Ensino de Ciências*, 1(3), pp. 193-232.
- Robins, A., Rountree, J. & Rountree, N. (2003). Learning and teaching programming: a review and discussion, *Computer Science Education*, 13(2), pp. 137-172.
- Rodríguez, T. (1999). *Teorías y modelos de enseñanza: posibilidades y límites*. Milenio Lleida.
- Romainville, M. (2000). Savoir comment apprendre suffit-il à mieux apprendre? En R. Pallascio, R. y L. Lafortune (dir.), *Pour une pensée réflexive en éducation*, pp. 71-86. Québec: Prensa de la Universidad de Québec.

- Shulman, L. S. (1986). Those who understand: knowledge growth in teaching. *Educational Researcher*, 15 (2), pp. 4-14.
- Solaz-Portóles, J & López, V. (2007). Resolución de problemas, modelos mentales e instrucción. *Revista Electrónica de Enseñanza de las Ciencias*, 6(1), pp. 70-89.
- Thurmond, V. (2001). The point of triangulation. *Journal of Nursing Scholarship*, 33(3), pp. 253-258.
- Varela, M. (2002). *La resolución de problemas en la enseñanza de las ciencias: aspectos didácticos y cognitivos*. Tesis para la obtención del título de doctor en Educación. Departamento de didáctica y organización escolar, Universidad Complutense de Madrid. Madrid, España.
- Villalobos, J. (2009). *Proyecto cupi2 – una solución integral al problema de enseñar y aprender a programar*. Bogotá: Universidad de los Andes.
- Villalobos, J. Casallas, R y Marcos, K. (2005). *El reto de diseñar un primer curso de programación de computadores*. Bogotá, Colombia: Universidad de los Andes, Departamento de Ingeniería de Sistemas.
- Vygotski, L. (1989). *El desarrollo de los procesos psicológicos superiores*. Barcelona: Grijalbo.
- Whalley, J. L., Lister, R., Thompson, E., Clear, T., Robins, P., Kumar, P. K. A. y Prasad, C. (2006). An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO taxonomies. Proceedings of the eighth Australasian computing education conference (ACE2006), (52), pp. 243-252.

Sobre los autores

Cristina Romero Chaves

Licenciada en educación especial, magíster en administración de la educativa y Candidata a Doctora en Ciencias de la Educación. Docente investigadora y Directora del grupo de investigación Modelos Pedagógicos (Anselmo Caradonna) de la Institución Universitaria CESMAG.
cristinaromeroch@gmail.com

María Mercedes Rosero Sosa

Trabajadora Social, Magister en Docencia Universitaria. Docente investigadora del grupo de investigación Modelos Pedagógicos (Anselmo Caradonna) de la Institución Universitaria CESMAG.
mamerosa@hotmail.com

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.