



LA DIDÁCTICA DE LA ENSEÑANZA DE LA MATEMÁTICA, ES POSIBLE APLICARLA A LA ENSEÑANZA DE LA PROGRAMACIÓN?*

Didactics in the Education of Mathematics ¿Is it Possible to Apply it to the Education of Programming?

Andrés Vargas García**

* Producto de diplomado en pedagogía, dictado en la Universidad Católica Popular del Risaralda, por el comité de pedagogía y currículo y la Vicerrectoría Académica.
** Ingeniero de sistemas y computación de la Universidad Tecnológica de Pereira, estudiante de Maestría en Ingeniería Eléctrica, línea ciencias computacionales, especialización en bioinformática de la Universidad Tecnológica de Pereira, docente Catedrático e investigador de la Universidad Católica Popular del Risaralda.



SINTESIS

Los avances tecnológicos han influido la vida en todos los aspectos; la educación no está exenta de este cambio, y las nuevas aplicaciones disponibles para escritorio, entornos web y dispositivos móviles han ayudado tanto a la enseñanza como al aprendizaje, pero cuando esta tecnología es usada para pasar por alto el proceso de aprendizaje, es donde deben emplearse nuevas técnicas, para captar la atención y motivar a los estudiantes de programación que ven este campo tan poco atractivo como para dedicar su vida a esto; es por ello que se analiza aquí cómo las didácticas que satisfactoriamente han usado el área de la matemática se pueden implementar en la enseñanza de la programación.

DESCRIPTORES:

Didácticas, matemáticas, informática, programación, TIC.

ABSTRACT

Technological progress has influenced our lives every way, education is not exempt from this change, and new applications available for both: desktop to the Web and mobile devices have helped both teaching and learning, but when technology is used to bypass the learning process that we must employ new techniques, or not so new, to engage and motivate students of programming, they see this field as unattractive to devote his life to this, is why we try to show that areas such as mathematics, have developed teaching them useful, which can implement the teaching of programming.

DESCRIPTORS:

Didactical, computer science, mathematics, pedagogy

LA DIDÁCTICA DE LA ENSEÑANZA DE LA MATEMÁTICA, ES POSIBLE APLICARLA A LA ENSEÑANZA DE LA PROGRAMACIÓN

Para citar este artículo: Vargas G., Andrés. (2010). "La Didáctica de la Enseñanza de la Matemática, es posible aplicarla a la Enseñanza de la Programación". En: Revista Académica e Institucional, Páginas de la UCPR, N° 88 p. 49-55.

Primera versión recibida: 17 de Noviembre de 2010. Versión final aprobada el 25 de Noviembre de 2010

Todo docente de cualquier programa del área de informática que tiene que enfrentarse al proceso de enseñanza de la programación, se ve en una encrucijada cuando su curso es de nivel medio o avanzado y sus estudiantes no han desarrollado una lógica algorítmica lo suficientemente fuerte para enfrentar problemas medianamente complejos. En este artículo trataremos de mostrar cuales didácticas ya conocidas, pueden ser útiles en este proceso.

No cabe duda de que las técnicas de enseñanza están evolucionando, la educación tradicional hasta hace muy pocos años era una transferencia de conocimientos desde el profesor hacia los estudiantes que escuchan atentamente, tratando de asimilar las ideas transmitidas; esta metodología de enseñanza expositiva tiene poca posibilidad de diálogo, porque en la mayoría de los casos los tiempos de clase son escasos para el desarrollo de temas complejos; y las clases prácticas, dictadas en laboratorios debido al número de estudiantes, no se prestan para eliminar el inconveniente de la interpretación de los conceptos, esto ocasiona que los estudiantes se sientan sin motivación, pues no han comprendido el tema desarrollado y no han participado de manera más fluida, perdiendo el interés en asistir a clase, y sólo lo hacen si existe un sistema obligatorio de asistencia.

Actualmente el uso de elementos multimedia, software de simulación, software didáctico de enseñanza, uso de las nuevas tecnologías de comunicación (TIC), web 2.0, han variado el concepto de enseñanza, transformando el ambiente rutinario por otro, caracterizado por la innovación y el cambio constante, pero eso no es suficiente para transmitir conceptos complejos en las aulas de clase, pues como estas nuevas herramientas ayudan al docente, también pueden hacer que el estudiante pase por alto el proceso de aprendizaje y facilite el copiado de tareas.

La Ingeniería de Sistemas requiere que el estudiante aprenda las metodologías existentes para el desarrollo de Software, y esto implica el aprendizaje tanto de

lenguajes de programación como de técnicas y conceptos que sustenten los estándares aplicados hoy en día para los procesos de desarrollo de software, y que además cuenten con metodologías y herramientas que les permita complementar lo aprendido en las aulas de clase, facilitándoles que en un momento determinado puedan actualizar sus conocimientos en forma autodidacta (Torres Méndez y Useche Zaldúa, 2006, p. 21).

En el proceso de formación de un profesional, debe pasar por múltiples etapas; el programa de ingeniería de sistemas y telecomunicaciones de la UCPR no es diferente, en las primeras etapas el estudiante se encuentra con una asignatura en donde el propósito es aprender «programación», y programar es la manera como se le da instrucciones a la computadora a través de la transcripción de un algoritmo a un lenguaje de programación, siguiendo alguna metodología y algún paradigma. Debido a la abstracción que se necesita para comprender los lenguajes de programación, los estudiantes ven esta materia como compleja y de difícil aprendizaje, creando una barrera que hay que romper para continuar con el proceso de enseñanza.

Existen discusiones sobre cuál lenguaje de programación se debe enseñar, cuál tecnología utilizar y que metodología seguir. Al respecto, los lenguajes de programación se dividen en (Alegsa 2007).

- **Paradigma Imperativo:** Describe la programación como una secuencia de instrucciones o comandos que cambian el estado de un programa. El código-máquina en general está basado en el paradigma imperativo. Su contrario es el paradigma declarativo. En este paradigma se incluye el paradigma procedimental (procedural).
- **Paradigma Declarativo:** No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe (declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo usado



para encontrar esa solución. Es más complicado de implementar que el paradigma imperativo, tiene desventajas en la eficiencia, pero ventajas en la solución de determinados problemas.

- **Paradigma Estructurado:** La programación se divide en bloques (procedimientos y funciones) que pueden o no comunicarse entre sí. Además, la programación se controla con secuencia, selección e iteración. Permite reutilizar código programado y otorga una mejor comprensión de la programación. Es contrario al paradigma in-estructurado, de poco uso, que es simplemente un «bloque», como por ejemplo, los archivos batch (.bat, Shell script).
- **Paradigma Orientado a Objetos:** Está basado en la idea de encapsular estado y operaciones en objetos. En general, la programación se resuelve comunicando dichos objetos a través de mensajes (programación orientada a mensajes). Se puede incluir -aunque no formalmente- dentro de este paradigma, el basado en objetos, que además posee herencia y subtipos entre objetos. Ej.: Simula, Smalltalk, C++, Java, Visual Basic .NET, etc. Su principal ventaja es la reutilización de códigos y su facilidad para pensar soluciones a determinados problemas.
- **Paradigma Funcional:** Concibe a la computación como la evaluación de funciones matemáticas y evita declarar y cambiar datos. En otras palabras, hace hincapié en la aplicación de las funciones y composición entre ellas, más que en los cambios de estados y la ejecución secuencial de comandos (como lo hace el paradigma procedimental). Permite resolver ciertos problemas de forma elegante.
- **Paradigma lógico:** Se basa en la definición de reglas lógicas para luego, a través de un motor de inferencias lógicas, responder preguntas planteadas al sistema y así resolver los problemas. Ej.: Prolog.

Otros paradigmas y sub-paradigmas son: paradigma orientado al sujeto, reflectante, basada en reglas, basado en restricciones, programación basada en prototipos, etc.

Se puede apreciar que es un campo complejo, pero en la formación de profesionales en sistemas informáticos se requiere el conocimiento de todos estos paradigmas.

La decisión de qué paradigmas usar se resuelve adoptando el criterio de demanda en el mercado, lo cual indica que la tendencia es la Programación Orientada a Objetos.

Ahora bien, para definir el lenguaje a utilizar debe interesar el análisis de sus particularidades, como la sintaxis, palabras reservadas, declaración de clases, asignación, entre otros.

Antes de llegar al lenguaje de programación primero debe entenderse el problema a resolver y crear un algoritmo en donde se describan los pasos a seguir para darle solución al problema; para esto se hace uso del lenguaje natural. Hay que tener en cuenta que el algoritmo de solución depende del paradigma elegido.

Éste algoritmo es creado en pseudocódigo o diagramas de flujo; de esta manera, un buen algoritmo luego se puede codificar en varios lenguajes, esta es una etapa previa a la codificación en un lenguaje de programación.

Y es aquí donde se centra el problema: la forma en que se enseña a los estudiantes como desarrollar un algoritmo para solucionar un determinado problema y luego realizar la traducción a un lenguaje en particular.

Los lenguajes de programación se vienen enseñando desde hace muy poco tiempo, en comparación con la enseñanza de la matemática, en donde se han creado y probado diversas metodologías y didácticas, aunque la discusión sobre si las matemáticas son un lenguaje es muy amplia e implica una compleja tarea en la que se combinan investigaciones matemáticas, conceptuales y empíricas (Quesada, 1991). A pesar de ello, es innegable que la matemática tiene un vocabulario especializado, una sintaxis particular y una semántica, y son estas características las que comparte con la programación; con base en esto puede sostenerse que la forma de enseñanza de las matemáticas sirve de referencia para la enseñanza de la programación.

Diferentes concepciones acerca de la naturaleza de la matemática y sus didácticas

En lo que al hacer matemático se refiere, algunos profesores lo asocian con la actividad de solucionar problemas, otros con el ordenar saberes matemáticos establecidos y otros con el construir nuevos saberes a partir de los ya conocidos, siguiendo reglas de la lógica (MEN 2003)

En general, consideran que la matemática en la escuela tienen un papel esencialmente instrumental, que por una parte se refleja en el desarrollo de habilidades y destrezas para resolver problemas de la vida práctica, para usar ágilmente el lenguaje simbólico, los procedimientos y algoritmos y, por otra, en el desarrollo del pensamiento lógico-formal.



• El constructivismo

El Constructivismo matemático es muy coherente con la Pedagogía Activa y se apoya en la Psicología Genética; se interesa por las condiciones en las cuales la mente realiza la construcción de los conceptos matemáticos, por la forma como los organiza en estructuras y por la aplicación que les da; todo ello tiene consecuencias inmediatas en el papel que juega el estudiante en la generación y desarrollo de sus conocimientos. No basta con que el maestro haya hecho las construcciones mentales; cada estudiante necesita a su vez realizarlas; en eso, nada ni nadie lo puede reemplazar.

• El Logicismo

Corriente de pensamiento que considera a la matemática como una rama de la Lógica, con vida propia, pero con el mismo origen y método, y que son parte de una disciplina universal que regiría todas las formas de argumentación. Propone definir los conceptos matemáticos mediante términos lógicos, y reducir los teoremas de la matemática, los teoremas de la Lógica, mediante el empleo de deducciones lógicas.

La Lógica era más un arte que una ciencia: un arte que cultiva la manera de operar válidamente con conceptos y proposiciones, un juego de preguntas y respuestas y un pasatiempo intelectual que se realizaba en la Academia de Platón y en el Liceo de Aristóteles.

Una de las tareas fundamentales del Logicismo es la “logificación” de la matemática, es decir, la reducción de los conceptos matemáticos a los conceptos lógicos. El primer paso fue la reducción o logificación del concepto de número. Según ello, la aritmética no sería más que una lógica más desarrollada; todo teorema aritmético sería una ley lógica aunque derivada. Las aplicaciones de la aritmética a la explicación de los fenómenos naturales serían un tratamiento lógico de los hechos observados; computación sería inferencia.

El conocimiento lógico-matemático

El conocimiento lógico-matemático no existe por sí mismo en la realidad. La fuente de este razonamiento está en el sujeto y él la construye por abstracción reflexiva; las operaciones lógico matemáticas, antes de ser una actitud puramente intelectual, requiere la construcción de estructuras internas y del manejo de ciertas nociones que son, ante todo, producto de la acción y relación del estudiante con objetos y sujetos y que, a partir de una

reflexión, le permite adquirir las nociones fundamentales de clasificación, seriación y la noción de número.

De acuerdo con las diferentes concepciones acerca de la naturaleza de la matemática y su didáctica, hacer matemáticas significa ocuparse de problemas, pero es igualmente importante en esta labor encontrar buenas preguntas, lo cual es totalmente compatible con la materia de programación en ingeniería de sistemas.

Es así como el profesor, tanto de matemáticas como de programación, debe crear un ambiente donde se propicien situaciones problemáticas que permitan al estudiante explorar problemas, construir estructuras, plantear preguntas y reflexionar sobre modelos; estimular representaciones informales y múltiples, y al mismo tiempo, propiciar gradualmente la adquisición de niveles superiores de formalización y abstracción; diseñar además situaciones que generen conflicto cognitivo teniendo en cuenta el diagnóstico de dificultades y los posibles errores. (MEN 2003)

Formas actuales de la enseñanza de la programación

En la actualidad existen dos enfoques principales al momento de enseñar a programar: uno usando la sintaxis y la semántica de un lenguaje, con la desventaja de que con la dinámica de la tecnología, un lenguaje puede extinguirse en cuestión de años y los estudiantes que solo aprendieron tácticas y trucos en este lenguaje quedarían desprovistos de herramientas. El otro enfoque es más algorítmico, que de manera general explica los conceptos de programación, y posteriormente permite su traducción a cualquier otro lenguaje; este método es más apropiado, pues cimienta bases y da herramientas al estudiante para enfrentarse no sólo a una tecnología en particular sino a múltiples lenguajes de un mismo paradigma, ya que, aunque se enseñe generalizando, cada lenguaje está estrechamente ligado a un paradigma de programación.

Por lo tanto, partiendo de la base que para la enseñanza de la programación se debe hacer mediante métodos algorítmicos, hace esto que nuestra mirada pedagógica se enfoque hacia la didáctica utilizada en la enseñanza de los mismos, logrando que los estudiantes se apropien del conocimiento; permitiéndoles resolver problemas en términos de un algoritmo para que posteriormente pueda ser llevado a un lenguaje de programación y ser implementado en una computadora probando su aplicabilidad, ya sea en programas, subprogramas, módulos, métodos o funciones.



La resolución de problemas no debe ser considerada solamente después de que haya ocurrido el aprendizaje, sino que ella puede y debe utilizarse como contexto dentro del cual tiene lugar el aprendizaje; por ello, esta visión exige al profesor que se creen situaciones problemáticas en las que los estudiantes puedan explorar problemas, plantear preguntas y reflexionar sobre modelos de resolución, logrando así entender la mecánica de la algorítmica. Los métodos tradicionales (tablero y papel) utilizados para el proceso enseñanza-aprendizaje se convierten en una tarea tediosa para el estudiante, logrando que se pierda interés en la clase y motivación por aprender; es aquí donde el profesor debe propiciar ambientes de aprendizaje que agreguen motivación al estudiante, ya que ella tiene impacto positivo sobre el aprendizaje.

El profesor debe buscar escenarios de aprendizaje lúdicos, donde el eje central de desarrollo de su proceso de enseñanza sea el juego, permitiendo que el estudiante estructure sus ideas, de orden y secuencia a sus acciones dentro del juego, logrando así resolver los problemas planteados; de esta forma, el proceso de aprendizaje pasa a ser activo en el cual se pueden ir incorporando los conceptos y habilidades necesarios para el diseño y construcción de algoritmos, que servirán de base al estudiante para la conversión del mismo en un lenguaje de programación.

Si durante el desarrollo de la clase se realizan actividades creativas y se dan los espacios suficientes para permitir al estudiante pensar, se logrará aumentar el grado de motivación, lo cual ayuda al estudiante a comprender los conceptos que de otra forma serían más difíciles de entender.

Es así como primero debe fortalecerse al estudiante en la solución de algoritmos, reforzando su lógica matemática, y ayudarle en el proceso de abstracción de la realidad al mundo algorítmico. Para esta tarea es de vital importancia conocer la estructura cognitiva del estudiante; no sólo se trata de saber la cantidad de información que posee, sino también cuáles son los conceptos y proposiciones que maneja, así como su grado de estabilidad, lo cual permitirá una mejor orientación de la labor educativa, esta ya no se verá como una tarea que deba desarrollarse desde cero, toda vez que los estudiantes tienen una serie de experiencias y conocimientos que afectan su aprendizaje y pueden ser aprovechados para su beneficio. Con actividades lúdicas, juegos, videojuegos, animaciones y demás medios se engancha y motiva a los estudiantes, para enfrentarlos posteriormente a un lenguaje propiamente dicho.

Para establecer un entorno de aprendizaje lógico-matemático se implementan los siguientes procesos:

- Utilizar diversas estrategias de interrogación.
- Plantear problemas de final abierto
- Construir problemas para conceptos claves
- Pronosticar y verificar los resultados lógicos
- Solicitar a los alumnos que justifiquen sus afirmaciones u opiniones
- Brindar oportunidad para la observación e investigación

Una buena metodología para la estimulación del pensamiento y el aprendizaje es uno de los programas de (Feuerstein 1999) denominado Aprendizaje mediado, en donde el educador se pone como intermediario entre el estudiante y el mundo, y permite que el mundo sea accesible para él.

Conclusiones

El proceso de enseñar a programar es un proceso complejo en el cual hay que tomar decisiones importantes sobre cómo transmitir conocimiento a los estudiantes: la metodología y didáctica a emplear es una elección significativa que suele dejarse para el final de las reflexiones en torno a la enseñanza-aprendizaje.

La capacidad para utilizar los números de manera efectiva y razonar de acuerdo con la lógica son expresiones de la Inteligencia Lógico-Matemática. Esta Inteligencia incluye la sensibilidad por los esquemas, las relaciones lógicas, los buenos razonamientos, las abstracciones y en general, lo que tiene que ver con el pensamiento matemático.

La creación de patrones, juegos y rompecabezas lógicos, el diseño e interpretación de gráficos, todo lo que ayuda al pensamiento científico y a todo pensamiento de nivel superior es propio de esta inteligencia, que en general poseen los matemáticos, geómetras, científicos, economistas e ingenieros informáticos.

Los proyectos de investigación iniciados adecuadamente desde los primeros años, el ejercicio del cálculo, la resolución de problemas de final abierto, el uso de los bloques constructivos del pensamiento y la planificación de estrategias son algunos de los muchos recursos que facilitan el desarrollo. (MEN 2006).

El área de la matemática se viene enseñando desde mucho antes que la programación y ha desarrollado didácticas útiles para transmitir los conceptos; empleadas



correctamente en el campo de la informática y más específicamente en la programación, estas didácticas obtendrían buenos resultados, pues el eje central de la programación es la algoritmia y está estrechamente relacionada con el pensamiento lógico-matemático.

Referencias

Alegsa. (2007). "Paradigmas de programación." from <http://www.alegsonline.com/art/13.php>.

Aparicio, J. A., Hoyos, O., y Niebles, R. (2004). De Velásquez a Dalí: las concepciones implícitas de los profesores sobre el aprendizaje. *Psicología desde el Caribe*. Universidad del Norte, 144-168.

Devincenzi, G. H., Di Rado, G. R., y Segovia González, L. A. (2004). Implementación de herramientas didácticas interactivas para la enseñanza de graduación en ingeniería. Brasil: Universidad Federal del Río Grande do Sul (UFRGDS).

Ferreira Szpiniak, A., y Rojo, G. A. (2006). Enseñanza de la programación. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*.

Feuerstein, R. S. K., Pnina; J.Tannenbaum, Abraham (1999). *Mediated learning experience (MLE)*.

Ministerio de Educación Nacional (7 de Junio de 1998). *Lineamientos Curriculares de Matemáticas*. Bogotá: Magisterio.

Ministerio de Educación Nacional. (2003). "Lineamientos curriculares." 2010, from <http://menweb.mineducacion.gov.co/lineamientos/matematicas/desarrollo.asp?id=4>.

M.N, M. M. D. P. R. (2006) "Las inteligencias múltiples. un aprendizaje diferente."

Quesada, D. (1991). ¿Es la matemática un lenguaje? *Revista de Filosofía*. [Revistas.ucm.es/index.php/RCSF/article/download](http://www.revistas.ucm.es/index.php/RCSF/article/download).

Torres Méndez, C. y Useche Zaldúa, G. A. (2006). Construcción de ayudas didácticas para la enseñanza de programación orientada a objetos usando Java™ 5.0. Bogotá: revista de Tecnología v.5 fasc. 2 p.97-102