
Santiago(131) mayo-agosto

Aproximación a la resolución de problemas de programación computacional y su lógica de algoritmización

MsC. Antonio Salgado-Castillo^I

Dra. Isabel Alonso-Berenguer^{II}

Dr. Alexander Gorina-Sánchez^{III}

*asalgado@csd.uo.edu.cu, ialonso@csd.uo.edu.cu,
gorina@contre.sum.uo.edu.cu*

Facultad de Matemática y Computación. Universidad de Oriente,
Santiago de Cuba, Cuba

Resumen

Se reflexiona sobre algunas contradicciones esenciales del proceso de formación del profesional de la Ciencia de la Computación e Ingeniería Informática, lo que permitió revelar la necesidad de desarrollar en ellos una cultura algorítmica, a partir de proponer una lógica –algorítmica para la resolución de problemas de programación computacional, sustentada en la relación entre la representación matemática y su generalización algorítmica, como alternativa didáctica para dar respuestas pertinentes a las demandas sociales en diversos contextos de actuación.

Palabras clave: cultura algorítmica, resolución de problemas, lógica algorítmica, programación computacional, contradicción.

Abstract

Reflections on some essential contradictions on the formation process of Computer Science and Information Engineering professionals, which allowed to reveal the need to develop an algorithmic culture, from proposing an algorithmic logic for solving computer programming problems, based on the relationship between the mathematical representation and its algorithmic generalization as a didactic alternative to give relevant response on social demands in various contexts of action.

521

Antonio Salgado Castillo, págs. 521-533

Keywords: algorithmic culture, troubleshooting, algorithmic logic, computer programming, contradiction.

Introducción

El proceso de informatización de la sociedad ha cobrado gran auge en los últimos tiempos al propiciar la aplicación de las Tecnologías de la Información y las Comunicaciones a las diferentes esferas y sectores de la sociedad, en aras de lograr una mayor eficacia y eficiencia mediante la optimización de recursos y el incremento de la productividad.

A esta realidad no escapan los profesionales de Licenciatura en Ciencia de la Computación e Ingeniería Informática, los cuales no solo necesitan apropiarse de los principales adelantos científico – técnicos relacionados con estas ciencias, sino también desarrollar habilidades para diseñar, escribir, depurar y mantener el código fuente de programas computacionales; código que debe ser escrito en un lenguaje específico y requiere de conocimientos de varias áreas, del dominio del lenguaje a utilizar, de algoritmos especializados y de la lógica formal.

Desde esta perspectiva de análisis, se revela la necesidad de desentrañar las insuficiencias del proceso de enseñanza-aprendizaje de la resolución de problemas de programación computacional en las carreras antes mencionadas para lograr una inserción efectiva de los egresados en su ejercicio profesional.

La repercusión social de ambas profesiones tiene su esencia en el desarrollo de procesos relacionados con los **sistemas informáticos en las organizaciones y en la solución de clases de problemas en dominios propios y de aplicación**, que les permitan obtener un incremento en la eficacia y la eficiencia de su funcionamiento, con técnicas para el análisis de los procesos computacionales, la información a procesar y las interrelaciones correspondientes. (Plan de Estudios D Ingeniería Informática y Ciencia de la Computación, 2007).

522

Por lo que el proceso en estudio, forma a los estudiantes como mediadores entre los procesos del ciclo de vida del sistema informático, la explotación de sistemas y herramientas de desarrollo; siendo capaz, de seleccionar y utilizar el equipamiento, técnicas y métodos más efectivos en la modelación y desarrollo de software para resolver problemas de la ciencia, la tecnología, la educación, entre otros.

En este sentido, se aprecian dificultades que tienen los egresados para comprender las situaciones problemáticas y modelarlas desde la programación computacional. Al respecto, un estudio realizado en la carrera de Licenciatura en Ciencia de la Computación de la Universidad de Oriente, permitió revelar las siguientes dificultades en la asignatura de Programación:

- Dificultades para comprender las situaciones problemáticas que se les plantean y modelarlas desde la programación.
- Limitaciones en la utilización de herramientas computacionales, lo que afecta el correcto diseño de los programas.
- Imprecisiones en las soluciones computacionales que se dan a las situaciones problemáticas, las cuales no siempre satisfacen las exigencias originales.
- Escasa destreza para desarrollar procesos de abstracción y decodificación de tareas socio – profesionales y situaciones problemáticas.

Estas insuficiencias en la apropiación de los contenidos de programación, han sido confirmadas a nivel internacional, y en tal sentido, los docentes – investigadores Ferreira, A. y Rojo, G (2005), de la Universidad Nacional de Río Cuarto, en Argentina, aseguran que los alumnos del primer año que cursan la asignatura Introducción a la Algorítmica y Programación presentan insuficiencias como: escasa destreza para desarrollar algoritmos de mediana o alta complejidad, falta de apropiación de una metodología de resolución de problemas e insuficiente experiencia en el manejo del lenguaje de programación utilizado en la asignatura.

Autores como Guibert, N, Guittet, L y Girard, P (2005) plantean que los estudiantes que se enfrentan por primera vez a la programación, presentan dificultades para desarrollar modelos viables para resolver el problema y describir una estrategia comprensible por la computadora o abstraer los diferentes comportamientos de una tarea en una única estrategia.

Por su parte, Davis, H.C, Carr, L, Cooke, E y White, S (2009) de la Universidad de Southampton, explican que en los cursos de Programación impartidos, los estudiantes no son capaces de usar correctamente los conceptos más comunes.

Santiago(131)2013

Por consiguiente, esta permanente búsqueda de "*una forma de enseñar a programar*", debe fundamentarse en procesos de análisis, interpretación y abstracción de la lógica y concepción del algoritmo, destacándose el uso de estos como recursos previos a la resolución de un problema de programación computacional.

Hablar de programación computacional, entonces, implica reconocerla como un proceso creativo y complejo que permita al alumno el desarrollo de sus potencialidades para utilizar un "conjunto de abstracciones interrelacionadas entre sí para la resolución de problemas" (Chesñevar, 2000: p.1).

De aquí que surja la necesidad de formar profesionales, capaces de desarrollar cualquier aplicación a través de la programación y poner esta en función de su especialidad como una herramienta integradora y sistematizadora de la actividad intelectual que se desarrolle en múltiples contextos socio – profesionales.

Por consiguiente, se plantea como **necesidad social** del presente trabajo: que los estudiantes de Ciencia de la Computación e Ingeniería Informática de la Universidad de Oriente, se apropien de una cultura algorítmica para dar respuestas pertinentes a problemas de programación computacional, en correspondencia con las demandas sociales en diversos contextos de actuación profesional.

Por lo que demostrar la necesidad del tratamiento científico de la formación de una lógica algorítmica, como alternativa de solución a la **contradicción social** existente entre la apropiación de contenidos de programación y su aplicación a la solución de problemas computacionales para potenciar un desempeño eficiente del profesional de la Ciencia de la Computación e Ingeniería Informática en la diversidad de contextos socio – profesionales, se asume como **objetivo** de este trabajo.

La apropiación de los contenidos de programación en la solución de problemas computacionales

524

Desde la perspectiva de la Ciencia de la Computación y la Informática, la actividad de programar requiere del "establecimiento de un conjunto de instrucciones ordenadas para que la computadora lleve a cabo una determinada tarea" (Remedios, 2006: p.2).

Antonio Salgado Castillo, págs.521-533

Desde el punto de vista computacional, resolver un problema implica el establecimiento de una sucesión de pasos elementales, donde cada uno genera un conocimiento nuevo, que se obtiene como inferencia lógica a partir de los conocimientos y experiencias del individuo y de las condiciones del problema o consecuencias derivadas de estas en pasos anteriores. La conjunción de estos pasos, entonces, permitirá fundamentar la exigencia del problema (Alonso, 2001).

Sin embargo, la complejidad que caracteriza a la programación y la actividad resolutoria hace que al llevarla a cabo se confronten serias dificultades, siendo una de las principales, la falta de éxito que tienen los estudiantes en el análisis y algoritmización de situaciones problemáticas en relación con su posterior implementación en un lenguaje determinado. Esto conduce, por consiguiente a que se diseñen e implementen programas en los que se reflejan diferentes inconsistencias:

- Puede suceder que el programa diseñado e implementado esté bien sintácticamente, pero no cumpla con la intencionalidad deseada; es decir, la parte semántica es incorrecta.

- Aun cuando el programa sea semánticamente correcto, puede ser que la solución que brinda no sea óptima, lo que obliga al programador a volver a la parte de análisis o comprensión.

- En ocasiones, el estudiante comprende, interpreta y representa correctamente la situación problemática que se le plantea y diseña correctamente el programa, pero se equivoca en la implementación de la parte sintáctica y semántica.

Todo esto conlleva a que la herramienta computacional no cumpla con los requerimientos exigidos, implicando un alto compromiso social. Si tenemos en cuenta que las ciencias informáticas tienen una alta incidencia en el desarrollo de software para la medicina, la economía, entre otras, un error semántico o sintáctico podría acarrear graves consecuencias al brindar un resultado que no sea correcto.

En ese mismo orden, otro de los obstáculos más comunes a los que se enfrentan los estudiantes durante el proceso de programación computacional, lo constituye el énfasis formativo que generalmente se da a la teoría de la programación enfocada a un lenguaje específico, subestimándose la necesidad de crear habilidades de algoritmización que sean eficaces para cualquier lenguaje.

Esto se traduce en que la resolución de problemas casi siempre es precedida por un proceso largo de instrucción formal donde al alumno se le enseña la sintaxis del lenguaje, la estructura de los programas y los comandos del compilador. Sin embargo, ante la diversidad de contextos socio – profesionales, el programador tiene que enfrentar tres problemas al mismo tiempo: encontrar una solución al ejercicio, dominar la sintaxis del lenguaje y utilizar eficientemente el compilador, intérprete o ambiente de programación. Esto conduce a que el estudiante emplee mucho tiempo en el aprendizaje de la sintaxis de un lenguaje y no desarrolle habilidades como programador que es lo que le permitirá desempeñarse profesionalmente.

Esto propicia que el estudiante no desarrolle un pensamiento algorítmico, necesario e imprescindible en todo profesional de las ciencias informáticas, sino que intente solucionar un problema usando como único recurso el conocimiento de una situación problemática similar. Si bien esto no es del todo errado, no siempre se tienen ejemplos similares, y ante esta situación el programador no cuenta con las habilidades necesarias para resolverla, lo que repercute directamente en el contexto socio – profesional donde se desempeña.

Otra de las dificultades más comunes en la práctica de la programación computacional está dada por la limitada comprensión del problema a resolver, etapa que en muchos casos es subestimada por los estudiantes. Siendo reconocida por numerosos didactas la importancia de tener bien claro el problema que se quiere resolver antes de proceder a su solución, pues muchas veces los alumnos fallan porque leen el problema rápido, de manera descuidada, sin poner atención a los detalles y sin preocuparse por comprenderlo completamente lo que los lleva a resolver un problema que es diferente del que se les plantea (Polya, 1965; Santos, 1993; Alonso, 2001).

526

Algo análogo sucede durante la siguiente etapa, es decir, durante la *concepción de un plan de solución*. En tal sentido, Whimbey (1993), reconoce que en ocasiones, el estudiante no tiene claro el procedimiento a utilizar y comienza a trabajar con el que más se le parece, sin razonar sobre su conveniencia, ni estimar los posibles resultados que debe obtener, resolviendo el problema de manera mecánica, sin reflexionar sobre la eficacia de lo que está haciendo.

Ello lleva a que, en esos casos, se pongan en práctica planes de solución que no están precedidos de un diseño pertinente y veraz, que garantice una lógica coherente para su solución.

Finalmente, con respecto a *examinar la solución obtenida*, este mismo autor menciona dentro de los errores más comunes, el llegar a la solución y no comprobar que sea la correcta, lo que se expresa en que generalmente se valida la solución con los datos que aparecen como ejemplo en el planteamiento del problema y si funciona adecuadamente se supone que el programa ya es correcto; pero en numerosas ocasiones, al probarlo con datos reales, no funciona.

Sin embargo, aunque la premisa fundamental sea privilegiar la enseñanza de la programación desde la algoritmización sobre los lenguajes de programación, todavía los enfoques didácticos actuales no van más allá de establecer una guía de lo que debe enseñarse en detrimento de cómo enseñarlo, por lo que se reduce el alcance formativo solo a modificar los currículos de la asignatura Programación, subvalorando el desarrollo de las habilidades y potencialidades necesarias en el estudiante para el diseño y construcción de los algoritmos a través de una lógica didáctica que les dé orden y claridad.

Ello, entonces, limita su impacto social y tecnológico ante la necesidad de perspectivas dirigidas a cómo desarrollar en los estudiantes sus potencialidades para "aprender a pensar algorítmicamente" para dar respuesta a la diversidad de problemas en su campo profesional.

Por ende, se requiere de la **apropiación de contenidos de programación** como expresión de la aprehensión de conocimientos, habilidades y valores para desarrollar procesos de formalización, modelación, construcción de algoritmos y programas para el análisis, diseño y realización de software. Esto implica preparar al estudiante en el empleo de la modelación matemático-computacional de diversos sistemas, para satisfacer las necesidades del área donde se desempeñe.

527

Por otra parte, su **aplicación a la solución de problemas computacionales** se interpreta como aquel proceso en el que no solo se trasladan, sino también se replantean las estructuras de contenido, se buscan nuevas relaciones y se reafirman, desde un

proceso de recreación y transformación en el proceso de apropiación del contenido, permitiendo la aplicación a nuevas situaciones y contextos profesionales, para desarrollar, aplicar y facilitar el uso de la capacidad potencial de las computadoras en el almacenamiento, recuperación y actualización de información. Además de realizar la programación más eficiente de modelos de representación de la información, determinando los paradigmas de programación adecuados, desarrollando y aplicando métodos de análisis para verificar la terminación y la corrección de las soluciones. También es capaz de analizar y evaluar rigurosamente las diversas estrategias de solución de problemas, con vistas a determinar la factibilidad de las soluciones algorítmicas, las estructuras de datos adecuadas que optimicen las estrategias de búsqueda de soluciones para problemas intratables, problemas no-bien definidos y/o con información parcial, como la mayoría de los problemas reales.

Lógica didáctica del proceso de algoritmización computacional: alternativa para responder a la resolución de problemas de programación

Reconocer el valor que tiene el desarrollo de una cultura algorítmica en el profesional de la Ciencia de la Computación e Ingeniería Informática supone reconocer la perspectiva interdisciplinaria de estas ciencias y su contribución social a la solución de problemas en otras áreas, lo que supone la formación de un profesional con las capacidades y habilidades de abstracción, formalización, modelación, construcción de algoritmos y su implementación, necesarias para la realización de su objeto de trabajo en una diversidad de contextos profesionales.

Propiciar esta formación, garantiza el desarrollo de un profesional con una cultura algorítmica que le permita elevar la eficacia y eficiencia resolutora como forma efectiva de prepararlos para un desenvolvimiento autónomo en su desempeño socio – profesional para dar respuestas pertinentes a los problemas de programación computacional desde sus modos de actuación profesional. Esto obedece a varias razones que remarcan la importancia de favorecer esta formación: un estudiante con la habilidad de algoritmizar es capaz de:

- Modelar e implementar matemática y computacionalmente soluciones a problemas reales del contexto donde se desempeñe.

Ø Diseñar programas computacionales basados en algoritmos independientes del lenguaje, lo que permite la generalización a diferentes problemas aun cuando no exista relación contextual entre lo que se quiere resolver y el objetivo primario para lo que se diseñó el algoritmo.

Ø Aprender a programar en cualquier ambiente de programación con mayor rapidez, pues teniendo la lógica de algoritmización el aprendizaje se reduce solo a las sintaxis de cada lenguaje.

Por consiguiente, las pretensiones de esta investigación, en aras de formar una cultura algorítmica en dichos profesionales, implica actuar desde nuestro campo de acción didáctico: la **dinámica del proceso de algoritmización computacional** haciéndose necesaria la búsqueda de nuevos procedimientos didácticos que favorezcan el perfeccionamiento del proceso en estudio.

Por otra parte, la relación contradictoria que se establece entre el manejo de los conceptos y lenguajes de programación por parte de los estudiantes y el insuficiente desarrollo de un pensamiento algorítmico en detrimento del proceso de resolución de problemas, constituyen elementos significativos que demuestran, no solo las limitadas potencialidades de dichos estudiantes para ampliar sus capacidades cognitivas en tanto les «permite establecer estrategias, construir algoritmos, analizar y comprender los programas escritos por otros y aprender la sintaxis de los lenguajes de programación» (Moroni, N y P. Señas, 2005: p.1), sino también el predominio de sesgos en los enfoques didácticos que abordan la resolución de problemas de programación, al no privilegiar la concepción del ordenamiento lógico que debe preceder la implementación computacional.

Por consiguiente, la orientación de esta dinámica formativa se debe constituir en el desarrollo del pensamiento algorítmico, sobre la base de potenciar en los estudiantes una cultura algorítmica que le permita elevar la eficiencia resolutora para dar respuestas pertinentes a problemas actuales de programación. Lo anterior tiene su expresión **en la elaboración de un sistema de procedimientos didácticos para la algoritmización computacional, sustentado en un modelo de la dinámica lógico-algorítmica para la resolución de problemas de programación computacional.**

El modelo mencionado se espera brinde mayores posibilidades de desempeño a los futuros egresados en la diversidad de contextos socio – profesionales, lo que implica articular en un movimiento didáctico único, la **modelación matemática** y su **sistematización algorítmica**, como eje integrador de la lógica algorítmica de la resolución de problemas de programación que se propone.

Desde esta perspectiva, se proveerá al estudiante, desde la dinámica de las Ciencias Pedagógicas, no solo de los procedimientos algorítmicos y heurísticos que le permitan elevar la eficacia y eficiencia resolutoria, sino también de elaborar una estructura representacional que incluya la secuencia de estados necesarios para alcanzar la meta, lo que se traduce en sus potencialidades para *modelar*, como habilidad medular en el proceso de resolución de un problema de programación computacional, desde su doble carácter: matemático y computacional.

Precisamente la lógica – algorítmica que se propone se sustenta en esta doble modelación: matemática y computacional, a partir de que la representación matemática de la situación problemática se concibe como el proceso que permite concretar, en un modelo matemático, la comprensión e interpretación realizadas, de manera que facilite la visualización y conceptualización matemática de los objetos y relaciones inherentes a la situación problemática analizada, desde la lógica esencial de esta ciencia. Esta representación matemática de la situación problemática se caracteriza por tener como contenido un conocimiento generalizado de dicha situación, de aquí que se pueda decir que tiene como condición una determinada abstracción, que toma en cuenta la situación en su esencia. Además, esta representación puede pasar por diferentes grados de generalización, esto hace que no se considere como una reproducción mecánica e invariable de la percepción de la situación, sino que se asuma como un proceso dinámico y variable. La relación entre la representación y la situación problemática regulará, entonces, las modificaciones a las cuales está sometida la primera.

530

Por otro lado, la representación matemática de la situación problemática permitirá la determinación y activación del conocimiento a emplear en su algoritmización y solución. De aquí que al comienzo del proceso, los conocimientos del individuo le permitirán obtener una representación matemática de la situación y, en consecuencia, serán recuperados de su memoria los procedimientos

concretos de algoritmización, si los conoce. En tal sentido, lo que guía la recuperación de los procedimientos apropiados, es la representación matemática que de la situación se forma dicho individuo; por ello se considera crucial dicha representación para tener éxito en la algoritmización, al ser la que determina qué conocimiento se activará en la memoria.

Esto permitirá concretar en un nivel de esencialidad matemática, la manera en que el estudiante se representa y construye su modelo mental, resultado de la abstracción realizada, como punto de partida para nuevas representaciones y abstracciones, en una búsqueda y resignificación constante de las relaciones lógico – matemáticas modeladas, como punto de partida para un nivel de esencialidad algorítmica.

Este proceso de algoritmización permitirá la resignificación y reestructuración de la representación matemática creada por el estudiante; a partir de una identificación e integración sistemática de estructuras computacionales, lo que conduce a una representación más general, expresada mediante pseudocódigos. Por consiguiente esta generalización trasciende el simple aprendizaje de las características de un lenguaje específico, encaminándose hacia el reconocimiento y aplicación del pseudocódigo, como herramienta base para la escritura del algoritmo, y cualidad que da cuenta del carácter generalizador del mismo.

Por consiguiente, la nueva lógica – algorítmica que se propone permite perfeccionar el proceso de resolución de problemas de programación computacional desde la dinámica de los procesos de enseñanza – aprendizaje a partir de formar profesionales competentes, lo que no se reduce solo a aplicar ciertos conocimientos informáticos. Deben formarse profesionales que estén aptos para activar sus propios recursos como gestores que generalizan la apropiación de sus recursos computacionales a la realidad social donde se insertan y la integren con otras ciencias para dar soluciones pertinentes a diversos problemas sociales y tecnológicos.

Revelar, por tanto, estas nuevas relaciones desde la dinámica de la lógica–algorítmica de la resolución de problemas de programación computacional, implica contribuir a dar respuesta a una necesidad social. Significa una nueva mirada, como respuesta concreta a una problemática que forma parte de nuestra realidad, la formación

continua de profesionales de las ciencias informáticas con las habilidades necesarias para enfrentar cualquier problemática desde su ciencia y en pos de desarrollar otras.

Conclusiones

Haber partido de las limitaciones y problemas en la formación profesional que presentan los estudiantes de las carreras mencionadas, permitió reconocer algunas contradicciones esenciales de este proceso pedagógico para dar respuesta a las exigencias sociales, lo que demanda la necesidad de potenciar un egresado con una sólida cultura algorítmica, lo que permitirá brindar soluciones óptimas a diversos problemas sociales y tecnológicos.

Este fenómeno se traduce en fuertes implicaciones sociales, pues muchos de los software o algoritmos desarrollados serán aplicados en los diferentes escenarios sociales (salud, educación, economía, etcétera), por lo que debe haber un fuerte compromiso por parte de dichos profesionales con la calidad de los mismos.

Desde estas reflexiones, se fundamenta la nueva propuesta de una lógica–algorítmica para la resolución de problemas de programación computacional, sustentada en la relación entre la *representación matemática* y su *generalización algorítmica*, como eje integrador que puede contribuir al desarrollo de habilidades de algoritmización en los estudiantes de ambas carreras, elevando así la eficiencia resolutora para dar respuestas a problemas de programación computacional.

Bibliografía

ALONSO, I. *La resolución de problemas matemáticos. Una alternativa didáctica centrada en la representación*. Tesis de Doctorado no publicada, Universidad de Oriente, Santiago de Cuba, Cuba.

CHESÑEVAR, C. (2000). *Utilización de los mapas conceptuales en la enseñanza de la programación*. Recuperado el 10 de diciembre, 2011 de <http://cs.uns.edu.ar/~cic/2000/2000-jornadas-mapas/2000-jornadas-mapas.pdf>

FERREIRA, A. y G. ROJO. Enseñanza de la programación. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*. Recuperado el 25 de junio, 2011 de <http://teyet-revista.info.unlp.edu.ar/numero-1.htm>

Antonio Salgado Castillo, págs.521-533

Santiago(131)2013

GUIBERT, N., GUITTET, L. y GIRARD, P. *A study of the efficiency of an alternative programming paradigm to teach the basics of programming*. Recuperado el 10 de enero, 2012 de <http://www.lisi.ensma.fr/fr/equipes/idd/publications.html>

MORONI, N. y SEÑAS, P. *Aplicación de mapas conceptuales hipermediales en la visualización de programas*. Recuperado el 22 de octubre, 2011 de <http://cmc.ihmc.us/papers/cmc2004-252.pdf>