

**LÓGICA ALGORÍTMICA PARA LA RESOLUCIÓN DE PROBLEMAS DE PROGRAMACIÓN COMPUTACIONAL: UNA PROPUESTA DIDÁCTICA**

LÓGICA ALGORÍTMICA PARA LA PROGRAMACIÓN COMPUTACIONAL

AUTORES: Antonio Salgado Castillo<sup>1</sup>Isabel Alonso Berenguer<sup>2</sup>Alexander Gorina Sánchez<sup>3</sup>Yaritza Tardo Fernández<sup>4</sup>DIRECCIÓN PARA CORRESPONDENCIA: Facultad de Matemática Computación. Universidad de Oriente. E-mail: [asalgado@csd.uo.edu.cu](mailto:asalgado@csd.uo.edu.cu)

Fecha de recepción: 31 - 07 - 2012

Fecha de aceptación: 28 - 12 - 2012

## RESUMEN

El carácter eminentemente social, cultural y tecnológico del proceso de resolución de problemas de programación computacional, junto a la complejidad y dificultades detectadas en su enseñanza, han contribuido a despertar la preocupación por el estudio de los procesos de comunicación, transmisión y comprensión de la Programación y a interesar a una amplia comunidad científica en correspondencia con el creciente desarrollo que ésta alcanza en la actualidad. Razón por la cual este trabajo tiene como objetivo que se develen, desde el punto de vista didáctico, los ejes integradores de una lógica algorítmica que sea contentiva de la solución a la contradicción que se revela en el proceso formativo entre la modelación matemática y su sistematización algorítmica, para potenciar un desempeño eficiente de los profesionales de la Ciencia de la Computación y de la Ingeniería Informática. En esta dirección se fundamenta una nueva propuesta didáctica consistente en una lógica algorítmica, en la que se precisan y explican aquellos procesos esenciales que deben de llevarse a cabo en la resolución de problemas de programación computacional. Desde la fundamentación teórica realizada, se concluye que estos procesos constituyen momentos didácticos necesarios para resolver coherentemente la contradicción antes señalada.

PALABRAS CLAVE: Lógica algorítmica, programación computacional, resolución de problemas

---

<sup>1</sup> Licenciado en Ciencia de la Computación. Doctorante en Ciencias Pedagógicas. Profesor Asistente. Departamento de Ciencia de la Computación. Facultad de Matemática y Computación. Universidad de Oriente. Cuba.

<sup>2</sup> Licenciada en Matemática. Doctora en Ciencias Pedagógicas. Profesora Titular. Departamento de Matemática. Facultad de Matemática y Computación. Universidad de Oriente. Cuba.

<sup>3</sup> Licenciado en Matemática. Doctor en Ciencias Pedagógicas. Profesor Auxiliar. Departamento de Contabilidad y Finanzas. Filial Universitaria Contra maestre. Universidad de Oriente. Cuba.

<sup>4</sup> Licenciada en Letras. Doctora en Ciencias Pedagógicas. Profesora Titular. Departamento de Idiomas. Facultad de Humanidades. Universidad de Oriente. Santiago de Cuba.

## **ALGORITHMIC LOGIC TO SOLVE COMPUTATIONAL PROGRAMMING PROBLEMS: A DIDACTIC PROPOSAL**

### ABSTRACT

The cultural, technological and eminently social character of the computer programming problems solving process, joined with the complexity and difficulties detected in their teaching, has contributed to increase the concern about the study of the processes of communication, transmission and understanding of computer programming and to attract the attention of a wide scientific community in correspondence with the growing development that this reaches at the present time. That is the reason why this paper has the objective of discover, from the didactic point of view, the integrators axes of an algorithmic logic that solves the contradiction that is revealed in the formative process between the mathematic modeling and their algorithmic systematization to empower an efficient performance of the professionals of Computer Science and Computer Engineering. In this sense a new didactic proposal is based, that consist in an algorithmic logic, in which are specified and explained those essentials processes that should be carry out to solve computer programming problems. Based on the theoretical fundaments, we concluded that these processes constitute didactics moments, required in order to solve the contradiction mentioned before.

**KEYWORDS:** Algorithmic Logic, computer programming, problem solving

### 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 (TIC) 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 en dichas esferas.

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

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 de Ciencia de la Computación y de Ingeniería Informática, y el impacto social de la formación de

los futuros egresados para una inserción efectiva de los mismos 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 en cualquier ámbito social, que les permitan obtener un incremento en la eficacia y la eficiencia de su funcionamiento, con técnicas para el análisis del entorno en función de delimitar los procesos computacionales, la información a procesar y las interrelaciones correspondientes. Así como la gestión de proyectos informáticos y computacionales con un alto nivel de profesionalidad, por lo que se reconoce que deben estar dotados de un conocimiento tecnológico, de organización, dirección de procesos y entidades, que le permitan desempeñarse en todos los sectores de la sociedad, como el objetivo que se persigue en su desempeño profesional (Ministerio de Educación Superior, 2007).

Por lo que el proceso que se estudia, contribuye a formar al Ingeniero Informático y al Licenciado en Ciencia de la Computación en su función de mediar entre los procesos del ciclo de vida del sistema informático, la explotación de sistemas y las herramientas de desarrollo, desempeñando diferentes roles en el equipo de trabajo, así como la gestión del conocimiento y la capacitación.

Consecuentemente, su actividad profesional se desarrolla en un amplio espectro de organizaciones sociales, a partir de su inserción de manera multidisciplinaria con especialistas de diversas ramas para concebir y desarrollar la solución informática que brinde respuesta a las necesidades del contexto, siendo capaz de asimilar los modelos correspondientes, seleccionar y utilizar el equipamiento necesario, así como las técnicas y métodos más efectivos para el procesamiento de la información, la modelación y desarrollo de sistemas informáticos capaces de resolver problemas interdisciplinarios relacionados con la solución por computadora en los diversos dominios de la ciencia, la tecnología, la educación, la producción y los servicios.

Precisamente en la resolución de los citados problemas interdisciplinarios, se aprecian las mayores dificultades que tienen los egresados de estas carreras. En tal sentido, un análisis realizado en la carrera de Licenciatura en Ciencia de la Computación de la Universidad de Oriente, a partir de informes de los últimos seis años sobre los resultados docentes de la asignatura Programación, permitió revelar las siguientes insuficiencias:

- Limitaciones en la comprensión de las situaciones problemáticas que se les plantean y en su respectiva modelación desde la programación.
- Selección y empleo inadecuado de estructuras computacionales que no permiten la verificación y validación de los algoritmos que se conciben y se implementan.

- Imprecisiones en las soluciones computacionales que se dan a las situaciones problémicas, las cuales no siempre satisfacen las exigencias originales.
- Escasas destrezas en la codificación de procedimientos computacionales en una diversidad de lenguajes computacionales.

Estas insuficiencias en la apropiación de los contenidos de programación han sido también confirmadas a nivel internacional. En tal sentido, Oviedo y Ortiz (2002) en un estudio hecho en la Academia de Computación de la PIICSA, México D.F, concluyeron que existe carencia de habilidades para programar y bajo aprovechamiento docente en esta asignatura.

Por otra parte los docentes – investigadores Ferreira y Rojo (2005), de la Universidad Nacional de Río Cuarto, en Argentina, aseguran que los alumnos del primer año que cursan la asignatura de Introducción a la Algorítmica y Programación, presentan insuficiencias, dentro de las que citan: 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.

Del mismo modo, autores como Guibert, Guittet y Girard (2005), plantean que los estudiantes que se enfrentan por primera vez a la programación en su proceso de formación, presentan problemas tales como: no logran desarrollar un modelo viable o estructura que permita resolver el problema, ni describir una estrategia comprensible para la computadora o abstraer los diferentes comportamientos de una tarea en una estrategia que los integre a todos. Además de constatar numerosas dificultades en la concepción de un programa por parte de estudiantes del primer año universitario, concluyendo que el aprendizaje para programar no puede reducirse a aprender la sintaxis de un lenguaje de programación, sino que el estudiante debe comprender la dificultad de esta tarea y construir un modelo correcto de cómo los programas se elaboran y ejecutan. Estos autores reconocen que la didáctica de la programación no puede sustentarse en enfoques esquemáticos orientados a aprender sólo lenguajes de programación, sino que ésta debe fundamentarse en procesos de análisis, interpretación y abstracción de su lógica y concepción algorítmica.

De manera similar los investigadores franceses Faouzia y Mostafa (2007), aseguran que la fragilidad del conocimiento de los estudiantes al programar se debe, en parte, a la falta de un modelo mental de la computadora, que le sirva de base para crear algoritmos viables. Sin embargo, si bien la concepción de dicho modelo mental pudiera considerarse una alternativa viable para favorecer el proceso de programación; ésta no es suficiente para crear programas eficientes, ya que deben incorporarse otros conocimientos matemáticos, lógicos y computacionales, que permitan una perspectiva más integradora para abordar dicho proceso.

En otro sentido, los investigadores cubanos González, Estrada, y Martínez (2006) proponen que los primeros conceptos de programación se enseñen a partir de la construcción de algoritmos, usando pseudocódigos, y reconoce que al hacerlo surgen dificultades con la codificación. Si bien se coincide con la dificultad planteada por estos autores, se considera no es significativa comparada con los problemas que acarrea el no enseñar a los estudiantes a construir los algoritmos, que en nuestra opinión son la base de la programación.

Por otro lado, las investigadoras argentinas Moroni y Señas (2005) afirman que la complejidad de los programas que se desarrollan actualmente exige del uso de técnicas efectivas de programación, para lo que debe ponerse énfasis en el diseño previo, con lo que connotan la utilización de los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema. Si bien estas autoras reconocen el valor del empleo de algoritmos como recurso previo a la resolución de un problema, su perspectiva de análisis no trasciende lo meramente declarativo, sin ofrecer posibles vías y procedimientos didácticos de cómo diseñarlos.

De igual forma, Whitfield y otros (2007), de la School of Computer Science of Liverpool Hope University, plantean que la resolución de un problema computacional es compleja en sí misma, requiriendo de habilidades tales como la identificación de los sub - problemas, el reconocimiento de relaciones, situaciones y modelos que permitan desarrollar un algoritmo para la solución y la traducción del mismo al código ejecutable. Con lo que remiten nuevamente a la necesidad de potenciar el proceso de algoritmización computacional desde una perspectiva didáctica, a través de la formación de habilidades para analizar, identificar, reconocer y modelar la situación problemática.

Una opinión generalizadora de lo complejo que resulta el proceso de enseñanza – aprendizaje de la programación, es la que ha sido emitida por Kaäsboll (2002), después de realizar un estudio en varias universidades del mundo, quien en sus informes asegura que a nivel mundial, entre el 25 y el 80% de estudiantes que en su primer año de universidad enfrentan cursos de programación, han desaprobado o han abandonado los estudios de esta materia.

Como se puede apreciar, todas estas investigaciones reconocen que el proceso de enseñanza – aprendizaje de la resolución de problemas de programación computacional tiene como centro de sus dificultades la algoritmización. Sin embargo, se quedan sólo a nivel del reconocimiento de dicha problemática, sin llegar a proponer soluciones didácticas orientadas a revelar la lógica del proceso de algoritmización computacional.

Del análisis hasta aquí realizado es necesario resaltar que la búsqueda de una forma de enseñar a programar debe fundamentarse en procesos de análisis, interpretación y abstracción de la lógica y concepción del algoritmo; con lo que se destaca el empleo de estos, como recursos previos a la resolución de un problema de programación computacional. En este sentido, se devela la

necesidad de potenciar en los profesionales de la Ciencia de la Computación y de la Ingeniería Informática, una formación que se centre en una lógica algorítmica que les permita proponer respuestas pertinentes a problemas actuales de programación, en correspondencia con las demandas sociales en una diversidad de contextos de actuación profesional.

Hablar de programación computacional, entonces, implica reconocerla como un proceso complejo y creativo, que ha sido abordado desde una diversidad de paradigmas y ha generado numerosas propuestas y discusiones en la búsqueda de una forma óptima de enseñar a programar, 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).

La sistematización epistemológica realizada anteriormente, ha podido revelar las limitadas propuestas teóricas y metodológicas existentes en el proceso de algoritmización computacional, las que aún no logran revelar las esencialidades didácticas del mismo, desde un sistema de relaciones que caracterice los ejes integradores de su lógica dinamizadora en la resolución de problemas de programación.

Es por ello, que con el propósito de elevar a niveles cualitativos superiores la actividad formativa del futuro egresado de las carreras de Licenciatura en Ciencia de la Computación y de Ingeniería Informática, el presente trabajo se trazó como objetivo principal develar, desde el punto de vista didáctico, los ejes integradores de una lógica algorítmica que sea contentiva de la solución a la contradicción que se revela en el proceso formativo entre la *modelación matemática* y su *sistematización algorítmica*, para potenciar un desempeño eficiente del mencionado egresado en una diversidad de contextos socio – profesionales.

## DESARROLLO

Una profundización en los procederes didácticos más relevantes del proceso de programación computacional, implica remitirse en primera instancia, al análisis de las particularidades distintivas de su proceso de enseñanza – aprendizaje y, en segundo lugar, al estudio de la didáctica de la resolución de problemas de programación computacional. Se considera a estos aspectos necesarios en cualquier propuesta didáctica sobre la lógica algorítmica para la resolución de problemas de programación computacional.

### 1. *El proceso de enseñanza – aprendizaje de la programación computacional*

Las diversas perspectivas para abordar la didáctica de la programación computacional se pueden estructurar para su análisis en dos variedades de propuestas: a partir de enfoques orientados a la especificidad de uno o varios lenguajes de programación haciendo énfasis en su sintaxis y su semántica (Chesñevar, 2000; Moroni y Señas, 2004; Soler y otros, 2008; Pérez, 2009); o bien, se orientan al empleo de un lenguaje algorítmico a través de pseudocódigos, lo bastante general cómo para permitir su traducción posterior

a cualquier lenguaje de programación (Salazar y Delrieux, 2004; Martínez, 2005; Ferreira y Rojo, 2005; González, Estrada y Martínez, 2006).

Algunos partidarios del primer enfoque proponen el uso de mapas conceptuales a través del diseño de un lenguaje de programación, que facilite la representación gráfica de los elementos estáticos y dinámicos de programas implementados en este lenguaje, así como el diseño, visualización y prueba de los algoritmos, Moroni y Señas (2004). Esta experiencia proporciona un complemento de inestimable valor en el proceso de enseñanza – aprendizaje de la programación, aumentando la comprensión de conceptos de un alto nivel de abstracción y complejidad.

En este mismo enfoque se reconoce como válida la propuesta hecha por Pérez (2009) de la Universidad Politécnica del Valle de México, consistente en un sistema informático – educativo que brinda un lenguaje de programación con ambiente visual denominado Nuevo Lenguaje de Programación (NLP), que usa a su vez seis lenguajes diferentes, con el objetivo de potenciar el desarrollo de algoritmos y la elaboración de programas y así reducir el tiempo para aprender a programar, planteando que mientras más se practique más se aprende. Este investigador afirma que se logran resultados alentadores con la herramienta, ya que los alumnos empiezan a descubrir la sintaxis de los seis lenguajes rápidamente y, por otro lado, si no se tiene experiencia previa en el desarrollo de la programación se adquiere a partir de que el software indica las correcciones sintácticas.

Sin embargo, aunque este parecería ser un proceder ideal, puede conducir al estudiante a una mecanización del pensamiento, ya que el mismo emplea mayor tiempo en aprender la sintaxis de los lenguajes, que en el diseño lógico del programa; por lo que esta contribución no se considera suficiente para el desarrollo de habilidades esenciales para la programación, tales como la modelación y la representación. Además, no tiene en cuenta las imprecisiones semánticas que reflejan el objetivo esencial del programa, siendo estas un indicador del cumplimiento de la intencionalidad del algoritmo. En resumen, aunque se reconoce la pertinencia de esta contribución, se considera que tampoco logra potenciar, desde una óptica integradora, las relaciones existentes entre la algoritmización y su posterior codificación en un lenguaje específico.

Por otro lado, los seguidores del segundo enfoque, desde una posición más comprometida con el desarrollo de habilidades de razonamiento lógico y modelación, proponen un modelo que deviene Programa Heurístico General para la enseñanza de la Programación (González, Estrada y Martínez, 2006), el que consta de las siguientes fases: detección del problema en la actividad cognoscitiva, confección de un modelo que refleje las características esenciales del fenómeno, expresión de este modelo en un sistema informático, puesta en práctica del mismo para resolver la problemática y, por último, su reformulación.

Esta propuesta se basa en un paradigma de programación específico: la Programación Orientada a Objetos (POO), que por definición está basado en un lenguaje de tercera generación o de alto nivel, que requiere de la formación de estrategias heurísticas particulares, resultando de vital importancia para la búsqueda de la solución a un problema. En tal sentido, la heurística juega un papel esencial en el diseño de nuevos algoritmos, la redefinición de estos, su optimización y su codificación.

Sin desdeñar esta propuesta, se considera criticable que se dirija a las generalidades de la programación computacional y no tenga en cuenta las esencialidades de la algoritmización. Además, soporta su estrategia en la enseñanza de la POO como vía para desarrollar la creatividad y el pensamiento heurístico del estudiante, sin tener en cuenta que la propia naturaleza de lo heurístico requiere, en primer lugar, un pensamiento algorítmico, el cual debe partir de lo simple para llegar a lo complejo. Por lo que comenzar a enseñar con la POO, cuya sintaxis es muy complicada, implicaría un tiempo considerable para su estudio, en detrimento del aprendizaje y desarrollo de habilidades inherentes a la algoritmización.

Dentro del enfoque didáctico que emplea pseudocódigos, se destaca la metodología de Ferreira y Rojo (2005), sustentada en la ejecución de varias fases para resolver un problema, en un lenguaje algorítmico estructurado de carácter general. Así, una vez obtenida la solución, se traduce a un lenguaje de alto nivel y se implementa siguiendo las etapas de: análisis, diseño, implementación y prueba. Sin embargo, aun cuando esta propuesta resulta válida, en tanto se reconoce a la algoritmización como parte importante del proceso de programación, no profundiza en las particularidades de los procesos que intervienen en la lógica didáctica de la misma, al no favorecer el componente lógico – matemático necesario para llegar a comprender y modelar el problema, previo al proceso de planteamiento en un lenguaje algorítmico estructurado.

Teniendo en cuenta lo analizado hasta aquí, podemos señalar que el carácter eminentemente social, cultural y tecnológico del proceso de resolución de problemas de programación computacional, junto a la complejidad y dificultades detectadas en su enseñanza, han contribuido a despertar la preocupación por el estudio de los procesos de comunicación, transmisión y comprensión de la Programación y a interesar, al respecto, a una amplia comunidad científica en correspondencia con el creciente desarrollo que ésta alcanza en la actualidad.

Consecuentemente, en aras de alcanzar el objetivo del presente trabajo, se hace necesario dirigir ahora el análisis hacia los elementos componentes de la resolución de problemas de programación computacional y sus aspectos didácticos más relevantes. Teniendo en cuenta que estos elementos constituyen un sustento de gran valor para el proceso de enseñanza – aprendizaje de la programación computacional.

## *2. Didáctica de la resolución de problemas de programación computacional*

Desde el punto de vista computacional, resolver un problema de programación implica el establecimiento de una sucesión de pasos elementales, cada uno de los cuales 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 éstas en pasos anteriores. La conjunción de estos pasos, entonces, permitirá fundamentar la exigencia del problema (Alonso, 2001).

En adición, la actividad de resolución de problemas de programación computacional requiere del “establecimiento de un conjunto de instrucciones ordenadas para que la computadora lleve a cabo una determinada tarea” (Remedios, 2006, p.2), de dónde se induce la necesidad de favorecer el desarrollo de las potencialidades en los estudiantes para diseñar, y mantener el código fuente de los programas computacionales.

Por sus beneficiosos efectos en el desarrollo de las capacidades cognitivas de los estudiantes, la programación enfocada a la resolución de problemas computacionales, suscita interés psicopedagógico, al contribuir al desarrollo de su actividad intelectual en tanto les “permite establecer planificaciones y estrategias, construir algoritmos, estructurar instrucciones, analizar y comprender los programas propios o escritos por otros, aprender la sintaxis de los distintos lenguajes de programación y comparar recursos usados por los programadores expertos y por los principiantes” (Moroni y Señas, 2005, p.1).

En tal sentido, el modelo que propone Polya (1965), aun cuando está pensado para problemas matemáticos y no se constituye en procedimientos estrictos e inalterables que puedan asegurar el éxito al resolver un problema, deviene alternativa viable para una orientación general, reconocida por su representatividad en la didáctica de la programación computacional, la cual se concreta en las siguientes etapas: comprender el problema, concebir un plan de solución, ejecutar el plan, y examinar la solución obtenida.

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 varias inconsistencias, entre las que podemos mencionar:

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

- Sucede con frecuencia que el estudiante comprende e interpreta correctamente la situación problémica que se le plantea y diseña correctamente el programa, pero en el momento de la implementación, por la complejidad del lenguaje de programación, se equivoca en la parte sintáctica y semántica.

Lo anterior tiene entre sus principales causas los limitados enfoques teóricos y didácticos aplicados a la enseñanza de la resolución de problemas de programación computacional, debido a perspectivas reduccionistas de abordaje mecánico de las estructuras de los programas, que no trascienden la presentación de gran número de ejemplos análogos, a lo que se integra el hecho de que aún no se privilegia la concepción del ordenamiento lógico que debe preceder la implementación computacional.

Esto propicia que el programador no desarrolle un pensamiento algorítmico, necesario e imprescindible en todo profesional de las ciencias informáticas, sino que se intente solucionar un problema usando como único recurso el conocimiento de una situación problémica 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.

Todo esto conlleva a que la herramienta computacional que se desea obtener no cumpla con los requerimientos exigidos lo que redundará en una demora en el tiempo de programación. Además de implicar un alto compromiso social, pues si tenemos en cuenta, que las ciencias informáticas tienen una alta incidencia en el desarrollo de herramientas para la medicina, la educación, la economía, entre otras, un error semántico o sintáctico, podría acarrear graves consecuencias en cualquiera de estas áreas, al brindar un resultado que no sea correcto.

En este mismo orden de ideas, otro de los obstáculos más comunes a los que se enfrentan los estudiantes durante la resolución de problemas 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 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, cuando llega el momento de poner a prueba sus conocimientos en la diversidad de tareas y contextos socio – profesionales, el programador novato tiene que enfrentar tres problemas al mismo tiempo: encontrar una solución al problema, dominar la sintaxis del lenguaje y utilizar eficientemente el compilador, intérprete o ambiente de programación (Ramírez, 1991). Estas insuficiencias conducen a que el estudiante emplee mucho tiempo, en el aprendizaje de la sintaxis de un lenguaje que puede ser compleja y no desarrolle habilidades como programador, que es lo que le permitirá

desempeñarse profesionalmente y dar solución a una diversidad de problemas sociales con la calidad y eficiencia requerida.

Otra de las dificultades más comunes en la resolución de problemas computacionales está dada por la limitada comprensión del planteamiento de dichos problemas, etapa que en muchos casos es subestimada por los estudiantes, motivo por el cual resultan comunes los errores durante dicho proceso de resolución.

Precisamente, ha sido reconocida por numerosos didactas la importancia de tener bien claro las exigencias del problema que se quiere resolver antes de proceder a su solución, lo que ha sido el resultado de las observaciones hechas al desempeño de los alumnos, que muchas veces 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; Pozo, 1994; Alonso, 2001).

Algo análogo sucede durante la siguiente etapa, es decir, durante la concepción de un plan de solución. En tal sentido Whimbey y Lochhead (1993) mencionan una serie de errores comunes de los estudiantes al resolver problemas, reconociendo 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 mucho sobre la eficacia de lo que está haciendo. Ello lleva a que, en estos casos, se pongan en práctica planes de solución que no están precedidos de un diseño pertinente y veraz, que garantice una lógica coherente para la continuidad del proceso.

Finalmente, con respecto a examinar la solución obtenida, Whimbey y Lochhead (1993) mencionan 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 prueba el programa con los datos que aparecen como ejemplo en el planteamiento del problema y, si funciona adecuadamente, se supone que el programa ya es correcto. Sin embargo, en numerosas ocasiones, al probarlo con ciertos datos reales, éste no se ejecuta.

De lo anterior, se puede inferir la gran complejidad que entraña el proceso de resolución de un problema de programación computacional, y la necesidad de desarrollar en los futuros profesionales de Licenciatura en Ciencia de la Computación y de Ingeniería Informática las habilidades para analizar, comprender, interpretar y modelar el problema a resolver. Creando algoritmos eficientes que permitan su codificación, siguiendo las reglas de un determinado lenguaje de programación, así como comprobando su eficacia para resolver la situación problemática original.

Aquí cabe destacar el relevante papel que juega la habilidad modelar, pues a criterio de estos autores, el proceso de resolución de un problema de programación computacional requiere de dos modelaciones, la primera de

carácter matemático, que según define Juan Raúl Delgado en su artículo “Las habilidades generales matemáticas” asocia a todo objeto real o ideal objetos matemáticos que representen determinados comportamientos, relaciones o características suyas (Delgado, 2000); y a partir de ésta, una segunda modelación de carácter computacional, más esencial aún que la primera. Por ello, será necesario dedicar especial atención al desarrollo de esta habilidad, en el proceso de enseñanza – aprendizaje de las asignaturas computacionales, especialmente en aquellas, cuyos objetivos esenciales estén encaminados a enseñar a programar.

También es preciso insistir, en que para tener éxito en la modelación, se debe lograr una correcta comprensión de la situación problemática que se aborda, considerando esa comprensión, según la definición de Wiltrock (1990, p.7), quien la expresa como “...una representación estructural o conceptualmente ordenada de las relaciones entre las partes de la información que se debe aprender y entre esa información y esas ideas y nuestra base de conocimientos y experiencias”.

De la anterior definición se puede inferir que la comprensión de una situación problemática depende, ante todo, de la representación que de la misma se haga la persona que trata de resolverla, y siendo así, la representación debe jugar un papel muy importante en su resolución, ya que el proceso comienza justamente con la concepción de una representación de la mencionada situación (Alonso, 2001).

A propósito de esta habilidad de representar, debe puntualizarse que la misma ha ido cobrando importancia en la última década, desde que la propuesta de Estándares Curriculares para el año 2000, del National Council of Teacher of Mathematics, la incorporó como uno de los diez Estándares de matemática. Estableciendo que “los programas deben enfatizar las representaciones matemáticas, para fomentar su comprensión, de modo que los estudiantes: elaboren y usen representaciones para organizar, registrar y comunicar ideas matemáticas; desarrollen un repertorio matemático que pueda ser utilizado intencional, flexible y apropiadamente; utilicen representaciones para modelar e interpretar fenómenos físicos, sociales y matemáticos” (Alonso, 2001, p.9).

Entonces, para la concepción de adecuadas representaciones matemáticas de la situación problemática bajo estudio, será necesario llevar a cabo un apropiado proceso de análisis de la misma, que conduzca a su correcta interpretación. Considerando la habilidad de interpretar como el proceso mediante el cual se atribuye significado a los objetos que conforman la situación, de modo que adquieran sentido en función de los conocimientos matemáticos o en función del fenómeno o problemática real de que se trate (Delgado, 2000).

Ahora bien, aunque la premisa fundamental sea privilegiar la enseñanza de la resolución de problemas computacionales desde la algoritmización sobre 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. Aspecto que no ha beneficiado el alcance formativo de los nuevos contenidos que se han introducido a través de modificaciones curriculares en la asignatura Programación, pues se subvalora 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 pertinente.

De aquí el interés por develar, desde el punto de vista didáctico, los ejes integradores de una lógica algorítmica para la resolución de problemas de programación.

### 3. *Didáctica de una lógica algorítmica para la resolución de problemas de programación computacional*

Debe partirse de reconocer el valor que tiene disponer de la didáctica de una lógica algorítmica que sirva de base para conducir la formación del proceso resolución de problemas de programación computacional en el profesional de la Ciencia de la Computación y de Ingeniería Informática. Supone la formación de un profesional con capacidades y habilidades de abstracción, formalización, modelación, construcción de algoritmos y su programación, necesarias para la realización de su objeto de trabajo en una diversidad de contextos profesionales.

Esta propuesta, 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 ejes integradores de la lógica algorítmica de la resolución de problemas de programación que se propone.

Siendo así, se brindará desde la dinámica de las Ciencias Pedagógicas, no sólo los procedimientos algorítmicos y heurísticos que permitirán elevar la eficacia y eficiencia resolutoria ante la diversidad de tareas y problemas profesionales, si no también una estructura representacional que incluye la secuencia de estados necesarios para alcanzar la meta, lo que se traduce en sus potencialidades para modelar; como habilidad medular del proceso de resolución de un problema de programación computacional.

Precisamente, la didáctica de la lógica algorítmica que se propone, parte de reconocer, que para resolver un problema de programación computacional se debe realizar una doble modelación: matemática y computacional.

La citada modelación matemática tiene como base la *representación matemática de la situación problémica*, que 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 problémica bajo análisis, desde la lógica esencial de las Ciencias Informáticas.

Por consiguiente, durante la modelación matemática se interpretará a la *comprensión del problema* a resolver como expresión de los sucesivos acercamientos que debe realizar el estudiante para comprender el significado de

una determinada situación problemática, a los efectos de su solución desde una lógica algorítmica. En esta dirección debe tenerse en cuenta que hasta que no se comprenda bien la situación que se aborda, todo el trabajo que se realice en aras de resolverla puede resultar inútil.

La comprensión implica un análisis de la situación problemática, es decir, su fragmentación en partes, para examinar detalladamente cada una, con el propósito de identificar los objetos reales, matemáticos o computacionales que las conforman, sus características y sus funciones, así como determinar las más importantes o esenciales a los efectos del objetivo que se persigue. Todo esto conducirá a una primera comprensión de la situación problemática, la que deberá irse perfeccionando a lo largo de todo el proceso de algoritmización.

Ahora bien, la comprensión de una situación problemática depende de los conocimientos que posee el estudiante (computacionales, matemáticos y sobre el mundo que le rodea), pues tendrá que ir creando, imaginando objetos y relaciones, en correspondencia con los objetos y relaciones externas que presenta la citada situación.

De aquí que los conocimientos y experiencias sobre el mundo sean sumamente importantes para lograr una adecuada comprensión de la situación, la presencia de conocimientos lingüísticos, por ejemplo, es obviamente necesaria, ya que el estudiante debe comprender el lenguaje y las experiencias por medio de las cuales recibe el planteamiento de la situación.

Si todos estos conocimientos resultan suficientes para procesar la información que brinda la situación, es muy probable que se pueda lograr una adecuada comprensión de la misma, la que facilitará su posterior solución. Aun así, durante la modelación matemática esta comprensión no es suficiente, si no se complementa con una *interpretación matemática del problema*, como expresión del proceso de reconstrucción matemática que realiza el estudiante, a partir de establecer asociaciones y relaciones entre los objetos que intervienen en la situación problemática, con lo que se atribuye un significado a dicho proceso en función de la situación o problemática real que se pretende resolver, desde un conocimiento matemático, adquiriendo con ello un sentido diferente y cualitativamente superior.

Aquí resulta de vital importancia este proceso encaminado a establecer asociaciones y relaciones, el que se da una vez que se comprende la situación y conduce a la realización de abstracciones sustentadas en objetos y relaciones matemáticas. Así, para establecer estas relaciones deberán conectarse los resultados de la exploración y vincular las diversas partes de la información obtenida, es decir; conectar los datos similares y localizar las diferencias y semejanzas, para que, una vez hecha la comparación, se puedan establecer los vínculos entre los componentes de la información y expresarlos matemáticamente.

Si se tiene en cuenta que el conocimiento matemático es un conjunto de esquemas mentales en los que el individuo ha organizado los conceptos,

principio, fórmulas y procedimientos de esta materia, se podrá entender que a través de esos esquemas es que el estudiante puede relacionar y organizar la información nueva, para interpretarla en forma significativa, en aras de resolver la situación problemática.

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 regula, entonces, las modificaciones a las cuales está sometida la primera.

En este momento, es que como parte del proceso de modelación matemática, 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.

Algo que no se puede perder de vista es que cada individuo crea sus representaciones matemáticas, de ahí que diferentes estudiantes pueden crear diferentes representaciones de una misma situación problemática. Esto más que una dificultad puede favorecer el proceso de enseñanza – aprendizaje de la algoritmización, siempre que sea bien aprovechado desde el punto de vista didáctico.

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, siendo expresión del carácter de integración y sistematización lógico – matemática, desplegado por el estudiante para desarrollar sus potencialidades en aras de comprender e interpretar pertinentemente la significatividad de los objetos y relaciones modelados. Aspecto que permite alcanzar estadios superiores en la representación matemática realizada durante el proceso resolutor, constituyendo un punto de partida para un nivel de esencialidad algorítmica.

De modo que, la sistematización algorítmica tiene como base a la *generalización de la representación matemática – algorítmica*, que se concibe como el proceso que permite expresar mediante pseudocódigos la relación existente entre la identificación de estructuras algorítmicas y la integración jerárquica de estructuras algorítmicas. Este proceso 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.

La *identificación de estructuras algorítmicas*, durante el proceso de sistematización algorítmica, es interpretada como aquel proceso mediante el cual el individuo reconoce las estructuras algorítmicas necesarias para garantizar una lógica coherente en el proceso de algoritmización computacional. Este reconocimiento se logra a partir del análisis de la representación matemática de la situación problemática, contentiva de los rasgos esenciales de dicha situación, en correspondencia con los conocimientos computacionales y matemáticos que posee el individuo.

Es decir, el individuo cuando ya tiene concebida una representación matemática de la situación problemática, tendrá que distinguir, dentro de un conjunto de estructuras algorítmicas conocidas por él, aquellas que le serán útiles para transformar los objetos y relaciones que aparecen en la citada representación, a partir del conocimiento de las características y funciones de dichas estructuras.

Por consiguiente, la identificación de estructuras lógico computacionales, presupone el dominio de ciertas habilidades y operaciones. No se puede identificar una estructura si no se ha aprendido a determinar las propiedades y funciones de la misma por medio de la comparación con otras estructuras, esto es lo que permite diferenciarlas. Pero determinar las propiedades y funciones de una estructura no es suficiente, debe saberse diferenciar las propiedades y funciones esenciales de las no esenciales, como habilidad clave para lograr la identificación. Esto requiere del dominio del concepto de propiedad y función, además de habilidades para diferenciar en las estructuras diversas propiedades y características.

Sin embargo, esta identificación de estructuras por sí sola no garantiza la concepción de un algoritmo que conduzca a la solución de la situación problemática que se aborda durante el proceso de sistematización algorítmica, y para lograrlo debe llevarse a cabo en estrecha relación con la *integración jerárquica de estructuras algorítmicas*, garantizando así la selección, análisis y concatenación adecuada de las estructuras, previamente identificadas por el individuo, para conformar el algoritmo que constituirá la estructura funcional del proceso de programación computacional.

Todo lo anterior implica identificar e integrar jerárquicamente las citadas estructuras para dar significado a las secuencias lógicas desde una interpretación algorítmico – computacional, lo que se constituye en vía estratégica para dirigir las operaciones y procedimientos individuales, a partir de un proceso de ordenación, disposición y estructuración algorítmica. Esto propicia una pertinencia de la intencionalidad lógico – computacional que lleva a garantizar la eficiencia algorítmica del proceso.

Es en este momento que se debe garantizar que la integración de las estructuras computacionales sea óptima, mediante el empleo de herramientas

de control, iterativas o de definición en el orden exacto. Esto permitirá que, cuando se implemente el algoritmo que se está diseñando, se haga un uso eficiente de la memoria del computador y que el tiempo de ejecución del mismo sea mínimo.

La optimización realizada por el estudiante contribuye al desarrollo de habilidades de programación, en lo referente al diseño y la concepción de algoritmos. De aquí que en esta etapa se connote el carácter integrador y desarrollador que posee la algoritmización como proceso computacional, capaz de propiciar en el educando capacidades cognitivas, al tener que integrar y optimizar simultáneamente.

Como resultado de la relación que se establece entre la de identificación e integración jerárquica de estructuras algorítmicas se configura, con un carácter de integralidad cualitativamente superior, la *generalización de la representación matemática – algorítmica*, permitiendo sintetizar y dinamizar la lógica de algoritmización computacional desplegada. Esta es expresión del proceso de 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.

Este proceso de 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, constituyéndose en una primera aproximación algorítmica de la vía de solución del problema computacional; cualidad que da cuenta, del carácter generalizador del mismo.

Así pues, la didáctica asociada a la nueva lógica algorítmica que se propone, está en condiciones de ser aplicada para perfeccionar la resolución de problemas computacionales a partir de diferentes instrumentos didácticos que posibiliten concretar sus ejes integradores: *modelación matemática* y *sistematización algorítmica*, mediante el desarrollo de una lógica algorítmica coherente con los procesos de comprensión del problema, interpretación matemática del problema, representación matemática de la situación problémica, identificación de estructuras algorítmicas, integración jerárquica de estructuras algorítmicas y generalización de la representación matemática – algorítmica, los cuales se constituyen en los momentos didácticos más importantes en los que se debe incidir para perfeccionar el proceso de enseñanza – aprendizaje de la programación computacional.

## CONCLUSIONES

La sistematización realizada a partir de los referentes teóricos – metodológicos del proceso de resolución de problemas de programación computacional, permitió revelar el limitado tratamiento que se le brinda actualmente a la lógica de algoritmización, pues no se concibe un proceso didáctico que permita

integrar en un movimiento único la modelación matemática y la sistematización algorítmica, como expresión de una lógica algorítmica pertinente.

Se fundamenta una nueva propuesta didáctica consistente en una lógica – algorítmica, en la que se precisan y explican aquellos procesos esenciales que deben de llevarse a cabo en la resolución de problemas de programación computacional. Desde la fundamentación teórica realizada, se concluye que estos procesos constituyen momentos didácticos necesarios para resolver coherentemente la contradicción que se establece entre los ejes integradores: modelación matemática y sistematización algorítmica. Por lo que se considera que dicha propuesta, una vez concretada en diferentes instrumentos didácticos, tiene potencialidad para contribuir significativamente al desarrollo de habilidades de algoritmización en los estudiantes de las carreras de Ciencia de la Computación y de Ingeniería Informática, dando respuesta a problemas actuales de programación computacional.

## BIBLIOGRAFÍA

Alonso, I. (2001). La resolución de problemas matemáticos. Una alternativa didáctica centrada en la representación. Tesis en Opción al Grado Científico de Doctor en Ciencias Pedagógicas, Universidad de Oriente, Santiago de Cuba, Cuba.

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

Delgado, J.R. (2000). Las habilidades generales matemáticas. Disponible en: <http://www.soarem.org.ar/Documentos/Actas%20de%20la%20VII%20Carem.pdf> [Consultado el 25 de junio de 2011]

Faouzia, B y Mostafa, H. (2007). Utilisation des NTICs pour l'apprentissage et l'autoévaluation de l'algorithme. SETIT 2007, 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications. TUNISIA, Marzo 25 - 29, 2007.

Ferreira, A. y Rojo, G. (2005). Enseñanza de la programación. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología. Disponible en: <http://teyet-revista.info.unlp.edu.ar/numero-1.htm> [Consultado el 25 de junio de 2011]

González, W., Estrada, V. y Martínez, M. (2006). Contribución al desarrollo de la creatividad a través de la enseñanza de la programación. Revista digital Pedagogía Universitaria. Vol 9. No.3. Disponible en: <http://169.158.24.166/texts/pd/1894/04/3/189404308.pdf> [Consultado el 12 de mayo de 2010]

Guibert, N., Guittet, L. y Girard, P. (2005). A study of the efficiency of an alternative programming paradigm to teach the basics of programming. Disponible en: <http://www.lisi.ensma.fr/fr/equipes/idd/publications.html> [Consultado el 10 de enero de 2012]

Kaasboll, J. (2002). Learning Programming. University of Oslo, 2002

Martínez, Y. (2005). En busca de una nueva forma de enseñar a programar. Investigación bibliográfica. Disponible en:

[http://www.mty.itesm.mx/rectoria/dda/riee/pdf05/27\(DTIE\).YolandaMtz.pdf](http://www.mty.itesm.mx/rectoria/dda/riee/pdf05/27(DTIE).YolandaMtz.pdf)

[Consultado el 4 de abril de 2012]

Ministerio de Educación Superior (2007). Modelo del Profesional de Ingeniería Informática y de Licenciatura en Ciencia de la Computación, Cuba, 2007.

Moroni, N. y Señas, P. (2004). Aplicación de mapas conceptuales hipermediales en la visualización de programas. Disponible en: <http://cmc.ihmc.us/papers/cmc2004-252.pdf> [Consultado el 7 de enero de 2011]

\_\_\_\_\_ (2005). Estrategia para la enseñanza de la programación. Disponible en: <http://cs.uns.edu.ar/jeitics2005/Trabajos/pdf/52.pdf> [Consultado el 4 de abril de 2012]

Oviedo, M. y Ortiz, F.G. (2002). La enseñanza de la programación. Disponible en: <http://bibliotecadigital.conevyt.org.mx/colecciones/documentos/somece2002/Grupo4/Oviedo.pdf> [Consultado el 25 de abril de 2012]

Pérez, R. (2009). Una herramienta y técnica para la enseñanza de la programación. Disponible en: <http://campusv.uaem.mx/cicos/imagenes/memorias/6tocicos2008/Articulos/Cartel%206.pdf> [Consultado el 13 de septiembre de 2010]

Polya, G. (1965). "Como plantear y Resolver problemas", Trillas, 1965.

Pozo, J.I. (1994). "La solución de Problemas", Santillana. Aula XXI, Madrid, 1994

Ramírez, R.V. (1991). NEWT, una herramienta de programación gráfica para la enseñanza del pensamiento algorítmico. IX Reunión de Intercambio de Experiencias en Estudios sobre Educación. Monterrey, N.L., México, Agosto de 1991

Remedios, M.A. (2006). La lógica de programación en los joven club de computación y electrónica. Disponible en: <http://www.monografias.com/trabajos41/joven-club-computacion/joven-club-computacion2.shtml> [Consultado el 20 de mayo de 2012]

Salazar, C. y Delrieux, C. (2004). Asignaturas introductorias a la programación: una discusión acerca de sus objetivos y contenidos programáticos. Disponible en: <http://lsm.dei.uc.pt/ribie/docfiles/txt20031212151824TCI12.pdf> [Consultado el 20 de febrero de 2011]

Soler, Y. y otros (2008). Mapa conceptual "tipos abstractos de datos" y sistema de visualización de programas SVP – SUBC: herramientas eficaces en la formación virtual del ingeniero informático. Congreso Virtual Iberoamericano de Calidad en Educación a Distancia. Disponible en: <http://es.scribd.com/doc/21739903/RD14.Pág.1-13> [Consultado el 10 de enero 2012]

Whimbey, A y Lochhead, J. "Comprender y Resolver Problemas", Visor Distribuciones. España. 1993.

Whitfield, A.K y otros (2007). Programming, disciplines and methods adopted at Liverpool Hope University. ITALICS Volume 6 Issue 4, October 2007 [ISSN: 1473-7507]

Wiltrock, R. (1990). Comprensión y representación. MacMillan Publishing Company.

