



**Encuentro Internacional de
Educación en Ingeniería ACOFI**

Innovación en las facultades de ingeniería:
el cambio para la competitividad y la sostenibilidad

Centro de Convenciones Cartagena de Indias

4 al 7 de octubre de 2016



EL ANÁLISIS Y LA SÍNTESIS EN LAS PRÁCTICAS NORMALES PARA OBTENER SOLUCIONES COMPUTACIONALES EN LA UNIVERSIDAD DE NARIÑO

Nelson Antonio Jaramillo Enríquez, Manuel Ernesto Bolaños Gonzales, Jesús Insuasti Portilla

**Universidad de Nariño
Pasto, Colombia**

Resumen

La elaboración de un programa computacional realizado como solución a un problema planteado, está determinado por la forma en que se estructura en la mente los conceptos de un procedimiento que le permite poner en práctica las actividades mentales de análisis y de síntesis necesarias para identificar ante el problema, los elementos de la solución, las relaciones entre estos y el orden lógico en que deben interactuar para llegar al programa solución.

La estructuración de estos conceptos, se puede dar en forma autodidáctica o en formación formal, como es el caso de la Universidad de Nariño que con los programas de Ingeniería de Sistemas, Licenciatura en Informática y Tecnología en Computación, prepara estudiantes en el campo de la computación para que planeen, analicen, diseñen e implementen soluciones computacionales, e incorpora en su plan de estudios las asignaturas y prácticas necesarias para estructurar en la mente de los estudiantes las competencias necesarias.

Desde este ámbito, el presente estudio hace una revisión de la forma como cada programa, estructura en los inicios de formación, las bases conceptuales que le permitan realizar las actividades mentales de análisis y de síntesis en torno a la construcción de una solución. De esta observación se concluye, que es determinante la forma como se desarrolla la cátedra, los elementos didácticos utilizados, el tiempo destinado a la teoría y la práctica, la utilización de materiales educativos computacionales o un lenguaje de programación para las prácticas, entre otros, en el desarrollo de las estructuras mentales que, según el estudio, asumen el análisis y la síntesis desde la heurística o desde lo algorítmica, o la combinación de ambos. Con la heurística los

estudiantes privilegian la experimentación y se mueven en el ensayo y el error para obtener un resultado, mientras que, con la algorítmica, se ajustan a pasos determinados que le aseguran una llegada a la solución.

Este estudio muestra las metodologías encontradas en la universidad, sus características y cómo afrontan las actividades de análisis y síntesis en la consecución de una solución computacional a un problema planteado.

Palabras clave: análisis y síntesis; heurística y algorítmica; programa computacional

Abstract

The development of a computer program developed as a solution to a specific problem is determined by the way the programmer structured in his/her mind the concepts of a procedure that allows to implement the necessary mental activities of analysis and synthesis to be identified in front of a stated problem, all elements that will participate in the presented solution, the relations between them and the logical order in which they must interact to reach the computer program solution.

The structuring of these concepts can be the result of various practices, from those obtained in a self-taught manner or those obtained in formal education, where a professional is formed to deal with problems that require computing solutions. It is the case of the University of Nariño with programs such as Systems Engineering, Bachelor of Computer Science and Computer Technology. These academic programs prepare future professionals in the field of computing to plan, analyze, design and implement computer solutions, for which each program assumes from their curriculum a syllabi that aims to structure in the minds of students the skills needed for these purposes.

From this scope, this study makes a thorough review of how each program in its curriculum, structure from the beginning the conceptual basis of a solution procedure for computational problems that allow students to perform mental activities related with analysis and synthesis around the construction of a solution. From this observation it is concluded that it is very determining how professors of programming fundamentals courses assume the development of his/her lectures, the didactic elements used, the time devoted to the theory and practice, the use of computer educational materials or a programming language for practices, among others, take students to develop mental structures to face a problem, as noted, they assume the most common analysis and synthesis from heuristics or from the algorithmic, or combination from both approaches. With the heuristic is observed that students favor experimentation and move on try-and-failure to get a result, otherwise the algorithmic approach fits a procedure or the specific steps that will ensure arrival at the solution.

This study will finally show the types of methodologies found in the University of Nariño, characteristics that identify them and how they favor the mental activities of analysis and synthesis in achieving a computational solution to the problem.

Keywords: *analysis and synthesis; heuristic and algorithmic; computer program*

1. Introducción

La Universidad de Nariño dentro de los programas de Ingeniería de Sistemas, Tecnología en Computación y Licenciatura en Informática busca formar profesionales capaces de proponer soluciones óptimas a problemas de sistemas y computación, los cuales en sus planes de estudio desde los primeros semestres introduce temáticas que pretenden llevar a los estudiantes a desarrollar competencias que les permita planear, analizar, diseñar y desarrollar soluciones óptimas a un problema planteado, en donde, las habilidades de análisis y de síntesis y el nivel que se alcance en el proceso de formación, son muy importantes al momento de entrar a detallar el procedimiento aplicado en la solución de un problema. Los fundamentos de programación en Ingeniería de Sistemas, Algoritmos en Tecnología en Computación y Programación I en Licenciatura en Informática, se presentan como la primera experiencia que tiene el estudiante de formarlas, y depende de la forma como se lo guíe para que se apropie de ellas, la que determina las características de análisis y de síntesis alcanzados.

Acorde a lo anterior y realizada una evaluación de los programas acerca del cumplimiento de esta expectativa, se encontró que cada programa asume el inicio de la formación en prácticas computacionales con diferentes metodologías, las que determinan en forma significativa en las prácticas necesarias en el planteamiento de una solución computacional a un problema planteado. Este estudio muestra que en Ingeniería de Sistemas en la asignatura Fundamentos de Programación se asume la práctica con materiales educativos computacionales como el Dfd [], en Tecnología en Computación en Algoritmos y Licenciatura en Informática en Programación I, se utiliza para la práctica el lenguaje de programación C#, aspecto que determina en las estructuras mentales que el estudiante relaciona según los cognoscitivistas en las teorías de procesamiento de la información (pozo) en la memoria permanente y que van a ser activadas de acuerdo al problema que se les plantee, en la memoria de trabajo que es la que realiza las diferentes acciones de la solución.

Con este conocimiento, los grupos de Investigación GRIAS y Galeras.Net en la línea, “Procesos educativos Apoyados por las nuevas tecnologías de la Información y la Comunicación”, preocupados por el desarrollo de buenas prácticas de programación desde los inicios de la formación y basados en las teorías de aprendizaje, buscan determinar las características mentales asociadas con el análisis y la síntesis que el estudiante desarrolla en las metodologías encontradas, que garanticen procesos de pensamiento que incluya operaciones mentales de descomposición de un todo en sus partes y la determinación de las relaciones entre estas (análisis) y la comprensión de cómo están organizadas las relaciones para obtener una construcción lógica (síntesis), necesarios al momento de dar la solución de un problema por computador.

Para nuestro trabajo y apoyados en los postulados de la teoría del procesamiento información, se busca entender para cada metodología encontrada, como es el tratamiento de la información en las diferentes memorias, como se estructura un nuevo concepto en estas, y como actúan en la memoria trabajo, y como su

comportamiento permite caracterizar e identificar los diferentes enfoques de comportamientos del estudiante en el desarrollo de un problema específico; es así como se observa, que las características del desarrollo de las metodologías llevan al estudiantes a asumir el análisis y la síntesis desde un enfoque heurístico o desde un enfoque algorítmico, o la combinación de los dos. Con la heurística los estudiantes privilegian la experimentación y se mueven en el ensayo y el error para obtener la identificación de los elementos, las relaciones y el orden lógico para obtener un resultado, mientras que con la algorítmica, se ajustan a un procedimiento, determinado por una serie de pasos lógicos que le aseguran en forma determinada la identificación de los elementos, las relaciones y por último la especificación de unos pasos lógicos para llegar a la solución.

Este estudio va a hacer una presentación de los fundamentos teóricos que la soportan, las características de los tipos de metodologías encontradas, los enfoques con que se asume el análisis y la síntesis en la consecución de una solución computacional a un problema planteado y las características que estos plantean en el manejo de la información, para terminar con las conclusiones.

2. Análisis y Síntesis

Los conceptos de análisis y síntesis se refieren a dos actividades complementarias en el estudio de realidades complejas. El análisis consiste en la separación de las partes de esas realidades hasta llegar a conocer sus elementos fundamentales y las relaciones que existen entre ellos. La síntesis, por otro lado, se refiere a la composición de un todo por reunión de sus partes o elementos. Esta construcción se puede realizar uniendo las partes, fusionándolas u organizándolas de diversas maneras (Bajo, M.T., 2004).

La capacidad de análisis y síntesis en la construcción de algoritmos nos permiten conocer más profundamente al problema con que nos enfrentamos, simplificar en un modelo descendente su descripción, descubrir de igual forma sus relaciones, para posteriormente establecer una construcción lógica de la solución manifestada en una serie de pasos lógicos que al ser recorridos dan la solución al problema planteado.

3. Métodos Heurístico y Algorítmico

Método algorítmico.

Consiste en aplicar adecuadamente una serie de pasos detallados que aseguran una solución correcta. Por lo general, cada algoritmo es específico de un dominio del conocimiento. La programación de computadores se apoya en este método.

Método heurístico.

El método consiste en generar candidatos de soluciones posibles de acuerdo a un patrón dado; luego los candidatos son sometidos a pruebas de acuerdo a un criterio que caracteriza a la solución. Si un candidato no

es aceptado, se genera otro; y los pasos dados con el candidato anterior no se consideran. La Programación heurística implica una forma de modelizar el problema en lo que respecta a la representación de su estructura, estrategias de búsqueda y métodos de resolución, que configuran el Paradigma heurístico.

Como diferencia un Algoritmo es un claro conjunto de instrucciones para resolver un problema, la heurística implica la utilización de un enfoque de aprendizaje y descubrimiento para llegar a una solución

4. Solución de problemas computacionales

Dentro del proceso de formación de los futuros profesionales en el campo de la programación se incorpora para este estudio a los autores Luis Joyanes y Osvaldo Cairó quienes plantean:

Según (Joyanes, 1999) “La principal razón para que las personas aprendan lenguajes y técnicas de programación es utilizar la computadora como una herramienta para resolver problemas. La resolución de problemas exige al menos los siguientes pasos: Definición, Análisis del problema, Diseño del algoritmo, Transformación del algoritmo en un programa, Ejecución y validación del programa”

Por otro lado (Cairó, 1995) plantea lo siguiente, “No existen reglas claras que nos permitan resolver un problema. Sin embargo, creemos que podemos ofrecer un conjunto de técnicas y herramientas metodológicas que permitan flexibilizar y estructurar el razonamiento en la solución de un problema. Eso provocará finalmente la construcción de algoritmos eficientes. Las etapas que debe seguir la construcción de un algoritmo son:” Problema, Análisis profundo del problema, Construcción del algoritmo, Verificación del algoritmo.

5. Material Educativo Computacional Dfd

DFD es un editor e intérprete de diagramas de flujo desarrollado en lenguaje C++. Permite editar, ejecutar y depurar algoritmos representados como diagramas de flujo. Fue pensado para la enseñanza de algoritmos básicos, pero se puede usar para construir algoritmos complejos usando recursión y arreglos de varias dimensiones. Desarrollado por Fabián Cárdenas Varela, Eduardo Daza Castillo y Nelson Castillo Izquierdo

6. Prácticas normales para obtener una solución computacional en la Universidad de Nariño

Después de realizar la respectiva observación se encuentra dentro de la Universidad de Nariño que los fundamentos de la lógica de programación en los diferentes programas, está determinado por la forma en que se lleva a la práctica los diferentes conocimientos asimilados, en donde se identifica las siguientes metodologías:

- Metodología con Material Educativo Computacional – Dfd. Realizada por lo los estudiantes de Ingeniería de Sistemas en la asignatura Fundamentos de Programación del primer semestre.
- Metodología con Lenguaje de Programación C#. Realizada por los estudiantes de Tecnología en Computación en la Asignatura Algoritmos del primer semestre y los estudiantes de Licenciatura en informática en la asignatura Programación I en el segundo semestre.

Las dos metodologías dentro de su desarrollo asumen un procedimiento algorítmico como el planteado por (Joyanes,1999) o (Cairó,1995) fundamentado en una serie de pasos que el estudiante debe abordar para llegar a la solución del problema, con la diferencia que con el Dfd tan solo se llega a la construcción del algoritmo y ejecución del mismo, mientras que con lenguaje de programación se asume además de la lógica de programación el lenguaje de programación C#.

7. Enfoques con que se abordan las actividades de análisis y de Síntesis

De las metodologías anteriormente identificadas se observa que el comportamiento de la actividad mental relacionadas con el análisis y la síntesis en la solución de un problema permite identificar claramente dos grupos, determinados específicamente por la forma en que se desarrolla la práctica del curso en cada uno de ellos. Estos son:

- Metodología con enfoque algorítmico-heurístico
- Metodología con enfoque algorítmico

7.1 Metodología con enfoque algorítmico y heurístico. En esta metodología predomina el enfoque algorítmico en donde el individuo trata de seguir una serie de pasos que le permiten llegar en a una solución. Pero combina también elementos heurísticos para abordar algunas etapas del diseño.

De acuerdo a lo observado en la universidad, Tecnología en Computación y Licenciatura en informática desarrollan este enfoque metodológico y presenta las siguientes características:

- El estudiante tiene una metodología claramente definida producto de un aprendizaje guiado por un experto en la solución de problemas computacionales. Los pasos a seguir corresponden a: Definición del problema, Análisis del problema, Construcción del algoritmo, Transformación del algoritmo en un programa, Ejecución y validación del programa.
- ¡El estudiante después de construir el algoritmo, realiza la codificación en lenguaje C#, realiza la compilación y ejecución. Apenas en este momento determina si la lógica funciona o no.
- El diseño de las entradas y salidas se realizan con un proceso de descubrimiento en donde se explora (ensayo y error) para obtener el resultado esperado, generalmente fuera de la lógica de construcción.
- El resultado total obtenido es un programa en Lenguaje C# y se le da mas énfasis es este resultado que al algoritmo.

7.1.1 Proceso de análisis

Identificación de elementos. Los elementos que maneja la memoria de trabajo corresponden a los conceptos propios del problema a resolver, la relación que existe entre estos, los conceptos propios de las instrucciones de un algoritmo y los conceptos que se relacionan con el lenguaje de programación C#, el cual maneja elementos sintácticos y ambiente de desarrollo que exige, gran esfuerzo mental,

Relación entre elementos: Las relaciones que se deben identificar deben entrar a manejar todos los elementos anteriormente descritos, los conceptos propios del problema, relacionados con la sintaxis que exige los algoritmos, ya sean en pseudocódigo o en diagrama de flujo y el lenguaje de programación.

7.1.2 Proceso de Síntesis

Construcción Lógica de la solución: La construcción relacional que es el resultado final, se resumen en un programa realizado en lenguaje C, el cual es el resultado de identificar una serie de pasos que involucran la construcción de un algoritmo en un primer momento, y el programa en lenguaje de programación en un segundo, lo que implica el manejo de una cantidad grande de elementos y relaciones.

7.2 Metodología con enfoque algorítmico. En esta metodología predomina que el estudiante sigue una serie de pasos claramente determinados que le permiten llegar a una solución.

De acuerdo a lo observado en la Universidad el programa de Ingeniería de Sistemas hace uso de esta metodología. Características:

- El estudiante tiene una metodología claramente definida para abordar la solución de los problemas, la cual consta de los siguientes pasos. Definición del problema, Análisis del problema, Construcción del algoritmo, Diseño del algoritmo, Codificación, Digitación, Ejecución y validación del programa
- Los pasos anteriores establecen que generalmente el individuo realiza en forma metódica todos los pasos que plantea el procedimiento hasta la construcción del algoritmo.
- El material educativo computacional (MEC) Dfd, le permite acceder al computador y mirar un resultado computacional sin necesidad de recurrir a un lenguaje de programación para hacerlo.

7.2.1 Proceso de análisis

- **Identificación de elementos.** Los elementos que maneja la memoria de trabajo corresponden a los conceptos propios del problema a resolver, los conceptos propios de las instrucciones de los algoritmos, en este caso diagramas de flujo de datos y el ambiente de desarrollo del programa Dfd.
- **Relaciones entre elementos:** Las relaciones que se deben identificar, hacen referencia a todos los elementos anteriormente descritos, las relaciones de los conceptos propios del problema, las relaciones de estos con la sintaxis que exige cada una de las instrucciones de los algoritmos en diagrama de flujo, las relaciones con el MEC Dfd que son las mismas.

7.2.2 Proceso de Síntesis

- **Construcción lógica de la Solución:** La construcción relacional que sigue una descomposición sistémica de arriba hacia abajo determinado por el problema específico a resolver, entrega un camino lógico para construir el algoritmo, el cual se lo puede probar directamente con el Dfd, lo que implica el manejo de una cantidad mucho menor de elementos y relaciones.

8. Conclusiones

- El ambiente en el cual se mueve el estudiante determina en gran medida las estructuras que se van a relacionar en la memoria permanente.
- El tener menos elementos dentro del ambiente de aprendizaje como en el caso de la utilización del MEC Dfd, permite que se dedique más tiempo al análisis y a la síntesis de un problema determinado antes de adentrarse al computador para probar su solución.
- Utilizar un MEC específico como el Dfd, permite al docente enfocarse única y exclusivamente en la construcción del algoritmo.
- La enseñanza de los inicios de la programación con un lenguaje de programación formal como el C#, necesariamente exige una dedicación del tiempo para su enseñanza y asumir una serie de pasos adicionales, lo que va en contra del proceso de análisis y síntesis del problema que se desarrolla.
- Se debe favorecer un ambiente donde la práctica se realice con herramientas genéricas como el Dfd que sea flexible a cambios tecnológicos y que aproveche en forma óptima todo el conocimiento.
- Desde el punto de vista de una construcción eficiente y eficaz de una solución a un problema específico se recomienda la utilización de MEC específicos en los inicios de la programación.

9. Referencias

- Cairó Osvaldo (1995) Metodología de la Programación. Computec. México DF. pp 5.
- Joyanes Luis (1999), Fundamentos de Programación. McGraw Hill/Interamericana de España S.A. España. pp 40
- Pozo Juan (1995). Teorías Cognitivas del Aprendizaje, Universidad Autónoma de Madrid 1995. POZO JUAN, Ignacio. Pág.119
- Salma Marisol Santos Silva . Diferencia entre método algorítmico y heurístico. Consultado el 20 de Junio del 2016 en http://www.academia.edu/15831318/DIFERENCIA_ENTRE_METODO_ALGORITMICO_Y_HEURISTICO
- Servicio de Innovación Educativa. Universidad Politécnica de Madrid. Consultado el 20 de Junio del 2016 en <http://innovacioneducativa.upm.es/competencias-genericas/formacionyevaluacion/analisis Sintesis>

Sobre los Autores

- **Nelson Antonio Jaramillo Enríquez:** Ingeniero de Sistemas, Especialista en Auditoría de Sistemas, Magister en Comunicación Educativa. Profesor tiempo completo. njaramillo@udenar.edu.co
- **Manuel Ernesto Bolaños Gonzalez:** Ingeniero de Sistemas, Especialista en Auditoria de Sistemas, Magister en Ingeniería. Profesor tiempo completo. mbolanosg@gmail.com
- **Jesús Insuasti Portilla:** Ingeniero de Sistemas, Magister en Docencia Universitaria, Magister en ingeniería de computadores con énfasis en sistemas informáticos en red. Profesor tiempo completo Jesus_insuasti@hotmail.com

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

Copyright © 2016 Asociación Colombiana de Facultades de Ingeniería (ACOFI)