

# “Estudio de la problemática relativa al uso de las sentencias secuenciales, selectivas e iterativas, en el diseño de algoritmos”

**Área de interés:** Computación Educativa

Isaías Pérez Pérez<sup>1</sup>, Silvia Soledad Moreno Gutiérrez<sup>2</sup>

<sup>1</sup>Universidad Autónoma del Estado de Hidalgo.  
Instituto de Ciencias Básicas e Ingeniería  
Carr. a Tulancingo s/n. Mineral de la Reforma, Hidalgo. México  
e-mail: isaiasp@uaeh.edu.mx, isaiaspp7@hotmail.com

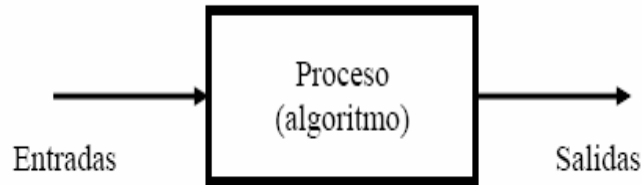
<sup>2</sup> Universidad Autónoma del Estado de Hidalgo  
Escuela Superior Tlahuelilpan  
Carr. Pachuca – Tula s/n. Tlahuelilpan, Hidalgo. México.  
e-mail: silviam@uaeh.edu.mx

**Resumen:** *El presente trabajo, basado en los resultados obtenidos en el estudio presentado por Pérez Pérez, et al. (2008), el cual mostró la evidente problemática presente en el desarrollo de la etapa de proceso en los algoritmos, por parte de los estudiantes de las asignaturas de programación estructurada en la UAEH. Para dar seguimiento a esto, se diseñó, aplicó y se llevo a cabo el análisis de un instrumento de evaluación, que se centra en abordar problemas donde se desarrollan “algoritmos lineales” (basados solo en sentencias secuenciales y selectivas), y en “algoritmos cíclicos” (que utilizan proposiciones secuenciales, selectivas e iterativas), con el propósito de identificar en que medida los estudiantes poseen los elementos conceptuales necesarios para el diseño de estos tipos de algoritmos.*

**Palabras clave:** *Diseño de algoritmos, Método, proceso.*

## Introducción

Hoy en día, dentro del denominado “Método de Resolución de Problemas por Computadora” (Joyanes Aguilar, 1993 y Watkins, 1984), la construcción de algoritmos se visualiza como una estructura sistemática clásica de entradas-proceso-salidas (ver figura 1).



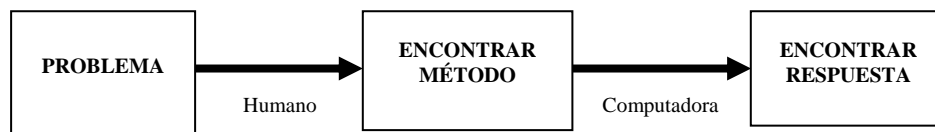
**Fig. 1. Componentes en el diseño de algoritmos, según el Método de Resolución de Problemas por Computadora.**

## Estado del arte y problemática presentada

Según Watkins (1984), en el campo de la resolución de problemas, se pueden visualizar tres grandes tipos de problemas:

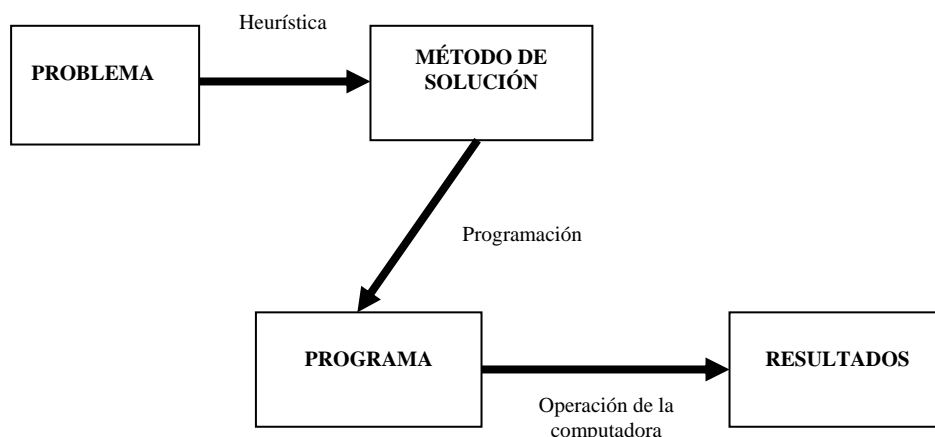
- a) *Los problemas que buscan respuestas.* Se trata de un trabajo en que se debe encontrar algo que se desconoce. La forma en que esto se haga no es de particular importancia siempre y cuando se obtenga la respuesta correcta (aunque es deseable que sea por un medio fácil).
- b) *Los problemas que buscan pruebas.* En estos, la tarea consiste en determinar la relación entre los datos y la respuesta, la cual ya se conoce. En este tipo de problemas, solo es necesario demostrar que existe o no una solución, sin crearla.
- c) *Los problemas que buscan métodos.* Aquí se busca un método mediante el cual se pueda derivar una respuesta. Este tipo de problemas son una parte de los problemas que buscan respuestas.

En la resolución de problemas mediante computadoras (ver figura 2), se trata de encontrar un método por medio del cual se pueda resolver un problema; dicho de otra manera, se busca diseñar un proceso del algoritmo, que después se codifica en algún lenguaje de programación específico. Una vez hecho esto, la computadora se hace cargo del mismo y suministra las respuestas solicitadas (ver figura 3).



**Fig. 2. Proceso general de resolución de problemas mediante computadora.**

Los problemas de computación no pueden ser problemas en que se busquen pruebas, porque el propósito del trabajo de las computadoras es encontrar respuestas que no se conocen de antemano. Pero tampoco se les puede considerar como problemas en que se busquen respuestas porque es la computadora misma la que determina la respuesta, no la persona que trabaja con ella. Hay que aclarar que no se esta discutiendo la forma en que las máquinas resuelven los problemas, sino como los resuelven las personas (Watkins, 1984).



**Fig. 3. Proceso detallado de resolución de problemas mediante computadora.**

Por tanto, la persona que desea utilizar una computadora debe ejecutar tres trabajos:

- a) Dado el problema, debe ser capaz de elaborar un método de resolución (heurística o resolución de problemas)
- b) Habiendo desarrollado un método, que se llamará algoritmo, se requiere convertirlo en una forma que la computadora pueda manejar (programación)
- c) Por ultimo, debe hacer que la computadora ejecute el programa para producir los resultados deseados y esto requiere un conocimiento del diseño de la máquina, de su organización y operación (Watkins, 1984).

El presente trabajo centra su interés sobre primer trabajo mencionado anteriormente, el del aspecto de la elaboración del método de solución o diseño del proceso del algoritmo. Según los resultados generados por la investigación de Pérez Pérez, et al. (2008), menos del 10% de los estudiantes de las asignaturas de diseño de algoritmos y programación estructurada, son capaces de llevar a cabo con éxito el diseño del proceso de los algoritmos.

## **Metodología o técnica usada**

La metodología que sigue el presente estudio es la siguiente:

- a) Selección de dos problemas tipo a plantear, definiendo cuidadosamente los enunciados de éstos
- b) Planteamiento de los enunciados de los problemas, a un grupo de estudiantes de la asignatura de programación estructurada
- c) Recolección de los resultados obtenidos
- d) Análisis y generación de las conclusiones, derivadas de los resultados obtenidos

Según el teorema estructural o teorema fundamental de la programación estructurada, cualquier programa, no importa el tipo de trabajo que ejecute, puede ser elaborado utilizando únicamente las 3 estructuras básicas: secuenciales, de selección y cíclicas o de iteración (Lozano R., 1986). Los dos problemas tipo que se escogieron fueron: el primero, que comprende solamente sentencias secuenciales y de selección (denominados algoritmos de tipo lineal) y el segundo, que requiere de los tres tipos de sentencias (llamados algoritmos cíclicos). La razón de haber escogido estos dos tipos de problemas, es porque reflejan diferentes niveles de dificultad en la construcción de los algoritmos, ya que los “algoritmos lineales” no requieren tener en cuenta elementos repetitivos, como es en el caso de los “algoritmos cíclicos”, además de que en éstos, se ha observado de manera empírica, que no son fáciles de desarrollar por parte de los estudiantes.

Cuando se plantean los enunciados de los dos tipos de problemas seleccionados a los estudiantes, se les solicita que lleven a cabo los tres pasos del diseño de algoritmos, fase propia del Método de Resolución de Problemas por Computadora (Joyanes Aguilar, 1993), los cuales son:

- a) *Diseño Descendente*, donde se desarrolla la secuencia de subproblemas, expresados de manera verbal, y que le permitan diseñar más adelante el algoritmo
- b) *Refinamiento de los Subproblemas*, donde se especifica de forma más abstracta o simbólica (de manera aritmética o algebraica), los subproblemas expresados de manera verbal anteriormente, y que se utilizarán para el diseño más detallado del algoritmo
- c) *Representación de Algoritmos*, en el cual se solicita que el estudiante represente el algoritmo, por alguna técnica como el diagrama de flujo, el pseudocódigo, las tablas N-S u otro que considere adecuado, basándose en la secuencia de subproblemas seleccionados en los dos pasos anteriores. Este último paso obliga al estudiante llevar a cabo una abstracción mucho más profunda del algoritmo a diseñar.

## **Resultados experimentales**

Los dos enunciados de los problemas seleccionados, se aplicaron a 21 estudiantes de la materia de programación estructurada, de primer semestre, de la carrera en Ingeniería en Electrónica y Telecomunicaciones, impartida en la Universidad Autónoma del Estado de Hidalgo. Después del análisis del desempeño de los estudiantes en dicha prueba, se obtuvieron los siguientes resultados (ver figura 4):

- a) En el caso del algoritmo lineal, el que usa solamente sentencias de tipo secuencial y de selección, los estudiantes tuvieron un éxito promedio de 69.8%, en el desarrollo de los pasos que comprende la fase de diseño de algoritmos, quedando evidente que conforme se avanza en los pasos desde el Diseño Descendente hasta la Representación de Algoritmos, el correcto desempeño de los estudiantes disminuye en poco más del 50%.
- b) En el caso del algoritmo cíclico, que utiliza los tres tipos de sentencias comprendidas dentro de la programación estructurada (secuenciales, selectivas e iterativas), el desempeño de los estudiantes, es significativamente menor que en el primer tipo de algoritmo. El promedio de éxito en el desarrollo de los pasos para el diseño del algoritmo fue apenas del 21.7%, poco más de tres veces menos que el desempeño que tuvieron los encuestados, en el desarrollo del algoritmo lineal.
- c) En términos generales, la tendencia observada en el diseño del algoritmo cíclico es similar a la del lineal (a la baja), pero con un descenso mucho más evidente, siendo este de más del 37%.

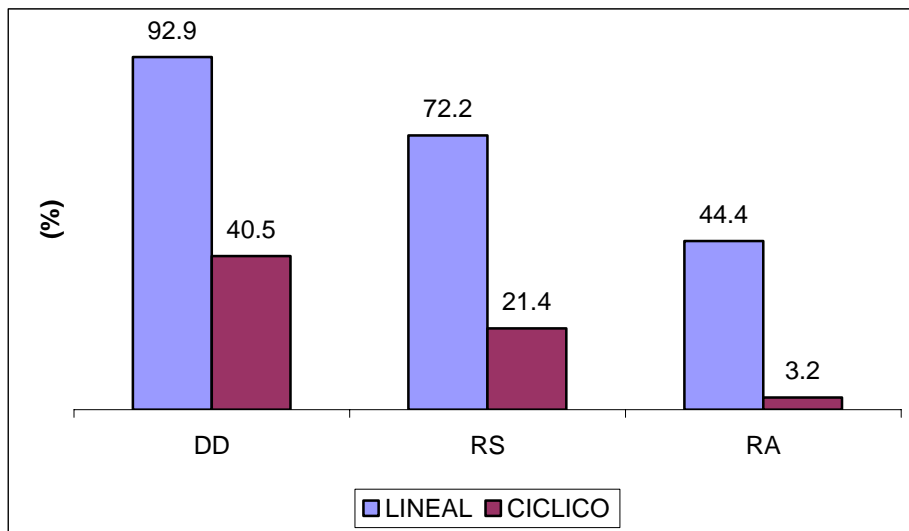


Fig. 4. Comparativa de los resultados del algoritmo lineal y el algoritmo cíclico.

## **Conclusiones y trabajos futuros**

Para comenzar, los resultados obtenidos en la presente investigación refuerzan los ya obtenidos en la investigación previa, mencionada anteriormente, como el hecho de que al avanzar en la especificación de los pasos de la fase del diseño del algoritmo (Diseño Descendente, Refinamiento de los Subproblemas y Representación de los Algoritmos), se refleja un claro descenso en el desempeño de los estudiantes, independientemente de que se trate de un algoritmo lineal o cíclico.

Haciendo un análisis preliminar, se puede decir que los pasos de Diseño Descendente, Refinamiento de los Subproblemas y Representación de los Algoritmos, muestran diferente nivel de especificación e interrelación entre las sentencias a utilizar. En Diseño Descendente, se identifican de manera verbal o con alguna simbología especial, las posibles sentencias necesarias para el diseño del algoritmo; en el Refinamiento de los Subproblemas, se busca representar de una manera más abstracta y simbólica las sentencias a utilizar, además de que se comienza a visualizar las posibles relaciones entre estas; y finalmente, en el paso de la Representación de los Algoritmos, al utilizar alguna técnica de representación de algoritmos, se debe comprender adecuadamente no solo las relaciones entre las sentencias a utilizar, sino además la dependencia entre unas y otras: es necesario definir una jerarquía entre ellas, poniendo en primer lugar las sentencias encargadas del manejo de la entrada de datos, pasando por las sentencias que llevan a cabo el proceso de cálculo o gestión de datos en el algoritmo, y finalmente, las sentencias encargadas de generar la salida de resultados.

Por otro lado, se puede vislumbrar otros aspectos en el diseño del proceso de los algoritmos, como el factor de utilizar o no sentencias secuenciales, selectivas o cíclicas, quedando evidencia que en el uso de estas últimas es donde los estudiantes tienen más problemas al tratar de implementarlas en el diseño de un algoritmo.

Las sentencias secuenciales y de selección son proposiciones que aunque controlan el flujo del algoritmos, este solo se da de un punto (inicio) a otro (fin), si que haya retrocesos o regresos en la secuencia de las instrucciones. En cambio, las sentencias cíclicas, manipulan el flujo del algoritmo de una manera más compleja, ya que las sentencias pueden empezar en un punto (entrada al ciclo), seguir un número de pasos específicos y según la condición lógica que controla el ciclo, este puede volver a empezar o terminar para seguir con la secuencia de normal del algoritmo. Las sentencias cíclicas, desde el punto de vista de como se estructuran las sentencias, se pueden concebir como submódulos, dentro del algoritmo total, ya que presentan algunas entradas al ciclo, un proceso iterativo y una salida del mismo.

Debido a todos estos hechos, se propone como siguiente paso, generar una hipótesis con bases teóricas robustas, nutrida de los elementos experimentales disponibles, que permita afrontar y dar solución de manera satisfactoria a esta problemática detectada.

## Referencias

Joyanes Aguilar, L. (1993). “*Fundamentos de programación. Algoritmos y estructuras de datos*”. Editorial Mc Graw-Hill. Primera edición. México.

Lozano, R., L. (1992). “*Diagramación y programación estructurada y libre*”. Editorial Mc Graw-Hill. Tercera edición. México. pp.: 264

Pérez Pérez, I., Fuentes Gálvez, A., Moreno Gutiérrez, S. S. (2008). “*Estudio de la problemática presente en el diseño de algoritmos por computadora*”. III Congreso Universitario de Tecnologías de Información y Comunicación 2008. Escuela Superior de Tlahuelilpan. UAEH. México.

Watkins, R. (1984). “*Solución de problemas por medio de computadoras*”. Editorial LIMUSA. Primera reimpresión. México. pp.: 9 a 11.