

# “Diseño de una propuesta de herramienta de diagramación para el desarrollo de algoritmos para computadora”

**Área de Conocimiento:** Computación Educativa

<sup>1</sup>Isaías Pérez Pérez, <sup>2</sup>Citlali Anahí Monzalvo López

<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: isaiaisp@uaeh.edu.mx, isaiaisp7@hotmail.com

<sup>2</sup>Universidad Autónoma del Estado de Hidalgo  
Sistema de Universidad Virtual  
Torres de Rectoría, 3er piso  
Carretera Pachuca-Actopan Km 4. C.P. 48900. Pachuca, Hidalgo. México.  
e-mail: cmonzalvo@uaeh.edu.mx, citlali.monzalvo@hotmail.com

## Resumen:

*El presente trabajo plantea el diseño de una propuesta de herramienta de diagramación, basada en la idea de los cursogramas, diagramas utilizados en la representación de diversos procesos. Dicha propuesta toma como base estructural el actual diseño sistémico de los algoritmos computacionales, de entrada-proceso-salida, así como también se toman en consideración las diversas observaciones que los estudiantes universitarios de computación han hecho al respecto, dentro de los estudios llevados a cabo por Pérez Pérez y Monzalvo López (2010 y 2012), y algunas observaciones relevantes mencionadas por Joyanes Aguilar (1988 y 1990) sobre el tema.*

**Palabras clave:** Algoritmos, cursograma, técnicas, herramienta, diagramación.

## Introducción

El planteamiento de un problema que pueda llegar a una solución, requiere la aplicación de una lógica que evolucione secuencialmente. Por otra parte, la resolución de todo problema exige tres grandes elementos: datos del problema, resultados solicitados y algoritmo de resolución.

Los datos del problema son información de partida y sobre la que normalmente no se puede actuar con la excepción de su manipulación correcta. Los resultados constituyen la información de salida y estarán íntimamente relacionados con la información de entrada. El algoritmo de

resolución es el conjunto de operaciones (matemáticas, lógicas, etc.) o manipulaciones que se deben realizar con los datos para llegar a la obtención de resultados (Joyanes Aguilar, 1988). Por tal motivo, el diseño y construcción de algoritmos para computadora se basan en la definición clásica de un sistema, el denominado diagrama de la “caja negra”: entrada-proceso-salida.

Es interesante observar, que los primeros métodos para desarrollo de algoritmos, basaban sus pasos en esta estructura tripartita de los algoritmos de computadora. Un ejemplo bastante representativo lo ofrece P. Morange (1975), cuando presenta el “Método empleado por el programador para preparar la escritura de las instrucciones de un programa”, para llevar a cabo el análisis del problema a nivel de una unidad de procesamiento o diseño de programas de computadora:

- 1) Descripción general del programa
- 2) Descripción de entradas y salidas. Se trata de describir la índole y el contenido de los elementos utilizados en la “entrada” para llevar a cabo el procesamiento y obtenidos a la “salida” como resultados de esa operación
- 3) Descripción del procesamiento. El programador aclara aquí la descripción del procesamiento.

## **Estado del arte y problemática afrontada**

Inicialmente, los algoritmos se suelen representar en forma narrativa, pero cuando tienen aplicación más directa, como es en el caso de la programación de computadoras, éstos se convierten en diagramas o gráficos de programación, que son la representación gráfica de la solución del problema que se desea mecanizar (Joyanes Aguilar, 1988).

Un diagrama de programación (o también llamado técnica o herramienta de diagramación), es la representación gráfica de los procedimientos y de la secuencia u orden en que deben ejecutarse éstos; en resumen, es la representación gráfica de la solución de un problema o de un procedimiento (Joyanes Aguilar, 1990).

Resultados obtenidos en los trabajos de Pérez Pérez y Moreno Gutiérrez (2008), demuestran que independientemente de la manera del planteamiento de los problemas en general, para los estudiantes universitarios de computación, les es más fácil determinar las salidas de resultados (en un 75%), que las entradas de datos (en un 60%) de los problemas planteados (Pérez Pérez y Moreno Gutiérrez, 2008). En definitiva, la entrada de datos afecta al proceso, y este afecta de manera natural a la salida de los resultados. El mayor porcentaje de errores cometidos por los estudiantes al tratar de definir un algoritmo de un problema específico, es en el diseño del proceso del algoritmo, en poco menos del 100% (Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez, 2008).

Para disminuir de manera sustancial este tipo de problemas, se desarrollo el llamado “Método de Resolución de Problemas por Computadora”, el cual entre sus técnicas se encuentran las denominadas herramientas de programación o técnicas de diagramación, como una forma de representar y finalmente, diseñar algoritmos para computadora. A pesar de esto, estas técnicas no han resuelto en su totalidad el problema, así lo declaran Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez (2008): “...Es necesario lograr enriquecer de manera teórica y práctica los procedimientos, técnicas y herramientas del actual Método de Resolución de Problemas por Computadora”.

Como punto de partida para proponer una mejora en el diseño de las herramientas de diagramación, se puede partir de las observaciones, ventajas, desventajas y comentarios más relevantes recogidos de las opiniones de los estudiantes, en las investigaciones de Pérez Pérez y Monzalvo López (2010 y 2012), sobre el uso, utilidad y aplicación de las herramientas de diagramación en el diseño de algoritmos:

- a) Permiten analizar los problemas de manera modular.
- b) El uso de las técnicas bien utilizadas, permiten generar más ideas de resolución de problemas. El poder hacer un diagrama indica que la persona es capaz de generar mentalmente la forma de solución del problema.
- c) Todas las técnicas son útiles para diseñar algoritmos; tienen sus puntos fuertes, así como sus desventajas. Usar varias técnicas de diagramación es recomendable para la comprensión del problema, así como para su solución. Para desarrollar buenos programas, se debería usar las tres técnicas de diagramación (diagramas de flujo, pseudocódigo y diagramas N-S), de manera secuencial o alternada, según se necesitara.
- d) Los diagramas permiten descubrir algún error de lógica de programa o una mejor forma de solución; por ejemplo, la tabla N-S puede funcionar como una prueba de escritorio.
- e) El uso de las técnicas de diagramación conocidas, en el diseño de algoritmos largos, complejos o cíclicos, es complicado, difícil de aplicar y consume demasiado tiempo. Los diagramas se podrían optimizar un poco, reduciendo las líneas o representando más información en menos espacio.
- f) Estas técnicas no ayudan generar diseños desde cero. Éste se debe concebir inicialmente en la mente del diseñador y las técnicas de diagramación sólo sirven para plasmarlo de manera gráfica, con el fin de aclarar y definir los detalles faltantes en el diseño.
- g) En las técnicas de diagramación no se especifican los tipos de datos a utilizar y eso puede causar conflictos en su implementación.
- h) Las técnicas de diagramación no permite llevar a cabo un método de prueba y error en el programa, como se puede hacer en un compilador, con el código fuente del algoritmo.

Otros aspectos a considerar al generar una propuesta de herramienta de diagramación, es que ésta debe combinar las ventajas de las técnicas conocidas, de una forma como lo hace los Diagramas de Nassi-Shneiderman (N-S): “Son una herramienta que reúne características gráficas propias de diagramas de flujo y características lingüísticas de los pseudocódigos” (Joyanes Aguilar, 1988, 1990).

Según Joyanes Aguilar (1990) y Pérez Pérez y Monzalvo López (2012), los estudiantes consideran con mayores ventajas de uso y didácticas al diagrama de flujo, seguido del pseudocódigo y finalmente al diagrama N-S; y en cuanto al mayor nivel de preferencia de uso, el pseudocódigo es el primero, seguido del diagrama de flujo y al último el diagrama N-S.

Después de llevar a cabo un análisis cualitativo de lo anterior, se puede generar una hipótesis como la siguiente: las herramientas de diagramación disponibles poseen ventajas y desventajas diversas; una idea inicial sería conjuntarlas de una manera en donde se potencialicen sus ventajas y disminuyan sus inconvenientes, cuando se utilicen para el diseño y construcción de algoritmos para computadora; además de que sigan la filosofía del diseño de entrada-proceso-salida, cosa que las herramientas de programación tradicionales no hacen hincapié en ello. Después de hacer una exploración documental exhaustiva, sobre como podría lograrse esto, se encontró la idea de los cursogramas.

Actualmente, los cursogramas se conceptualizan como tablas que representan gráficamente los diversos componentes de información que transitan por las diversas partes de un sistema durante cierto proceso realizado. El cursograma es una técnica gráfica desarrollada con la finalidad de describir operaciones de algún proceso en general, y entre esos, pueden representarse procesos de computadora (Chapin, 1977). Por esta misma razón, en los años 70's, se les consideró incluso sinónimos de los diagramas de flujo, debido a sus enormes similitudes, aunque hoy en la actualidad, ambas representaciones han tomado caminos separados, pero no por eso no son herramientas gráficas interesantes y dignas de tomar en cuenta. Los cursogramas son fáciles de aplicar y de aprender, pues se basan en unas pocas reglas por demás sencillas y comprende escasos componentes. Sirven para describir sin ambigüedad los procesos que sufren los datos dentro de la computadora. También son adecuados para representar las operaciones previas y posteriores a los procesos internos (entradas y salidas) (Chapin, 1977).

Además, se puede observar que en el uso de los diagramas de flujo, utilizados en sistemas, contienen elementos gráficos (o símbolos) comunes con los símbolos de los cursogramas; las diferencias principales son las diferencias de significado de los símbolos además de que no todos se utilizan en unos y en otros (Folgar, 1999). Por todas las razones y similitudes mencionadas, la propuesta a presentar, se basará en diversos aspectos de los actuales cursogramas, adecuados a un proceso del diseño de un algoritmo para computadora.

## **Descripción de la metodología o técnica usada**

Basándose en la utilidad de los cursogramas, se propone el siguiente método para desarrollar una propuesta de una nueva técnica de diagramación:

- 1) Investigar sobre las generalidades de los cursogramas, las ventajas e inconvenientes que presentan, así como las pautas y técnicas generales que se siguen para su confección.
- 2) Analizar las observaciones recabadas de los estudiantes y las recomendaciones de expertos relativas al tema, con el fin de tener una serie de elementos conceptuales a incluir en la propuesta.
- 3) Generar una propuesta concreta de cursograma, en donde se incluya el uso de las diversas herramientas de diagramación, así como la estructura de entrada-proceso-salida de los algoritmos, aspectos de la declaración de tipos de datos y recomendaciones diversas de los expertos y los estudiantes.
- 4) Dar a conocer la propuesta a través de ejemplos específicos.

## **Resultados experimentales**

Después de hacer una profunda investigación y un minucioso análisis de la construcción de los cursogramas, se generó la siguiente “Técnica para el desarrollo de cursogramas de algoritmos para computadora”, basándose, principalmente, en algunas consideraciones mencionadas por Folgar (1999):

- a) Diseñar un diagrama tabular, con la identificación del algoritmo, declaración de tipos de datos, un inicio y fin, así como los rubros de entrada, proceso y salida respectivos (ver figura 1).
- b) Esquematar o representar, utilizando las técnicas de diagramación conocidas (diagramas de flujo y pseudocódigo), de la forma más simple, la estructura del algoritmo a desarrollar, pero distribuyendo en el rubro que le corresponda (entrada-proceso-salida), cada instrucción que integrará al algoritmo a diseñar. Si es necesario, es posible incluir comentarios en cualquier parte del diagrama para mejor documentación del mismo.
- c) Revisar paso a paso lo expresado en el diagrama, para verificar si la secuencia y estructura es correcta o no. Es recomendable, hacer una prueba de escritorio exhaustiva.
- d) Llenar el rubro de declaración de tipos de datos, con los identificadores (constantes y variables) requeridos en el diagrama.
- e) Hacer la versión final del cursograma, tomando en cuenta todos los aspectos y modificaciones contempladas en su confección.

## **Conclusiones y trabajos futuros de investigación**

Se puede decir finalmente, que se tiene el supuesto de que siguiendo la técnica propuesta, se podrán desarrollar algoritmos que representarán de manera más adecuada, la estructura natural de entrada-proceso-salida de los algoritmos, además de que esta

también intenta aprovechar las ventajas de las herramientas de diagramación como el diagrama de flujo y el pseudocódigo, lo cual repercute en una mejor aceptación y utilización de la herramienta, por parte de los estudiantes. Es necesario dar a conocer dicha propuesta y hacer las pruebas respectivas, para tener idea clara si la propuesta representa un avance en el mejoramiento de las técnicas de diagramación disponibles.

<b>CURSOGRAMA DE ALGORITMO PARA COMPUTADORA</b>		
<b>Nombre:</b>		
<b>Comentarios:</b>		
<b>Declaración de tipos de datos:</b>		
<b>ENTRADA</b>	<b>PROCESO</b>	<b>SALIDA</b>
<b>INICIO</b>		
		<b>FIN</b>

**Fig.1. Esquema general de la propuesta de cursograma de algoritmo para computadora.**

## Referencias

- Chapin, N. (1977). *“Cursogramas”*. Primera edición. Editorial “El Ateneo”. Argentina. p. 7.
- Folgar, O. F. (1999). *“Los procedimientos. Cursogramas, diagramas de proceso y formularios”*. Primera edición. Ediciones Macchi. Argentina. p. 54,55, 61 a 63.
- Joyanes Aguilar, L. (1988). *“Metodología de la programación. Diagramas de flujo, algoritmos y programación estructurada”*. Primera edición. Editorial Mc Graw-Hill. México.
- Joyanes Aguilar, L. (1990). *“Problemas de la metodología de la programación”*. Primera edición. Editorial McGraw-Hill. México. p. 6, 7, 10, 21, 23.
- Morange, P. (1975). *“Introducción a la programación”*. Segunda edición. Editorial “El Ateneo”. Argentina. p.37 a 41.
- Pérez Pérez, I.; Moreno Gutiérrez, S. S. (2008). *“Estudio sobre la problemática en los enunciados de los problemas de programación”*. III Congreso Estatal Universitario en Tecnologías de Información y Comunicaciones 2008. Instituto de Ciencias Básicas e Ingenierías. UAEH. México.
- 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.
- Pérez Pérez, I.; Monzalvo Lopez, C. A. (2010). *“Análisis del desempeño de los estudiantes en el diseño y construcción de algoritmos secuenciales, selectivos e iterativos”* y *“Nivel de utilización de las técnicas de diagramación por parte de los estudiantes, en el diseño de algoritmos secuenciales, selectivos e iterativos”*. V Congreso Universitario en Tecnologías de Información y Comunicaciones 2010. CITIS. ICBI. UAEH. México.
- Pérez Pérez, I.; Monzalvo López, C. A. (2012). *“Análisis del nivel de utilización de las herramientas de diagramación, manejadas dentro del diseño de algoritmos”*. En prensa. VII Congreso Universitario en Tecnologías de Información y Comunicaciones 2012. Área académica de Computación. ICBI-UAEH. México.