

# “Análisis del nivel de utilización de las herramientas de diagramación manejadas dentro del diseño de algoritmos”

**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: isaiaasp@uaeh.edu.mx, isaiaasp7@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, que se basa en los resultados de las investigaciones de Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez (2008) y Pérez Pérez y Monzalvo López (2010), exhibe los resultados obtenidos de un estudio de tipo mixto, donde se abordan aspectos cualitativos para después cuantificarlos, sobre el tema de la utilización de las diversas técnicas de diagramación disponibles (diagrama de flujo, pseudocódigo y diagramas N-S), empleadas en la expresión de algoritmos para computadora. Todo esto es con el propósito de determinar cuales de las herramientas de diagramación son las que tienen más repercusión dentro del desarrollo de algoritmos, por parte de estudiantes de la carrera de Sistemas Computacionales, impartida en la Universidad Autónoma del Estado de Hidalgo.*

**Palabras clave:** Diagrama, uso, algoritmo.

## **Introducción**

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; de ahí que “la estructura de un algoritmo se puede representar en un diagrama estructurado en forma de bloques donde se muestren las diferentes tareas que deben ser ejecutadas y su relación entre ellas” (Joyanes Aguilar, 1990).

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

Actualmente, los tres tipos fundamentales de herramientas de diagramación más utilizadas, dentro de los lenguajes algorítmicos son: los Diagramas de Flujo, el Pseudocódigo y los Diagramas N-S:

*Diagramas de flujo.* Es una serie de símbolos, cada uno de los cuales representa una función en el programa que se encuentra conectada a la siguiente en dirección descendente por las líneas de flujo. Es un medio de presentación visual y gráfica del flujo de datos, a través de un algoritmo. El diagrama de flujo representa realmente a un sistema de proceso de información, las operaciones ejecutadas dentro del sistema y la secuencia en que se ejecutan.

Los diagramas de flujo han sido la herramienta de programación por excelencia y, aún hoy, siguen siendo muy empleados. Son fáciles de diseñar porque el flujo lógico del algoritmo se muestra en un dibujo en lugar de con palabras. Sin embargo, los diagramas de flujo tienen una limitación grande: son difíciles de actualizar y complejos de leer si ocupan varias páginas; cuando se han de modificar los diagramas de flujo resultan difíciles precisamente por su carácter gráfico (Joyanes Aguilar, 1988, 1990).

*Pseudocódigo.* Es una mezcla de lenguaje natural y símbolos, términos y otras características comúnmente utilizadas en uno o más lenguajes de alto nivel. Nació como un sustituto del lenguaje de programación, y por consiguiente sus palabras reservadas se conservaron o fueron muy similares a las de dichos lenguajes. El pseudocódigo es un lenguaje algorítmico similar al español o al inglés pero más conciso y que permite una redacción rápida del algoritmo aunque lógicamente no tendrá la precisión en sus operaciones que los lenguajes de programación. El pseudocódigo es fácil de mantener, ya que sus modificaciones se pueden realizar incluso con procesadores de texto.

El pseudocódigo es una técnica para expresar en lenguaje natural la lógica de un programa, es decir, su flujo de control. El pseudocódigo no es un lenguaje de programación sino un modo de plantear un proceso de forma que su traducción a un lenguaje de alto nivel sea sencillo para un programador. La fase de confección de pseudocódigos es inmediatamente anterior a su codificación en el lenguaje de programación elegido. En la actualidad el pseudocódigo es una técnica muy utilizada sobre todo en la programación de lenguajes estructurados como Pascal, o bien utilizando técnicas de programación estructurada. Una de sus principales ventajas es que no existe un conjunto de reglas que definan con precisión lo que es y lo que no es el pseudocódigo, ya que varía de un programador a otro. El pseudocódigo surgió como alternativa al diagrama de flujo, ya que

supera las dos principales desventajas del diagrama de flujo, que es lento de crear y difícil de modificar sin un nuevo redibujo. (Joyanes Aguilar, 1988, 1990).

*Diagramas de Nassi-Shneiderman (N-S).* Son una herramienta que reúne características gráficas propias de diagramas de flujo y características lingüísticas de los pseudocódigos. De manera natural, es una herramienta de programación que favorece el diseño de algoritmos siguiendo la programación estructurada.

Un diagrama N-S es como un diagrama de flujo con las flechas omitidas y cajas o bloques contiguos. Las acciones sucesivas se escriben en cajas sucesivas. Como en un diagrama de flujo, diferentes acciones pueden ser escritas en una caja. El diagrama se lee de arriba-abajo. Cada bloque ejecuta una operación específica que se puede documentar o describir con la precisión que se desee. La diferencia entre esta forma y las otras dos formas (diagrama de flujo y pseudocódigo) parece trivial. Sin embargo, esto es debido a que los algoritmos son bastante sencillos. Este sistema de representación permite tener una visión mucho más estructurada de ellos y por consiguiente mayor facilidad al traducirlos al lenguaje de una computadora. Estos diagramas tratan de optimizar los programas, ya que permiten gran flexibilidad, al permitir fáciles correcciones, modificaciones o ampliaciones al diagrama original (Joyanes Aguilar, 1988, 1990).

En definitiva, las técnicas de diagramación como los diagramas de flujo, el pseudocódigo y los diagramas N-S, son esquemas representativos, además de un lenguaje para expresar algoritmos; además de que el aprender a utilizar estas técnicas de diagramación es relativamente sencillo; lo que resulta más difícil es aprender a utilizarlas como herramientas efectivas para expresar ideas. Esto llega sólo con la práctica (Forsythe, Skeenan, Organick, Stenberg, 1975).

A este respecto, los trabajos de Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez (2008), muestran que los estudiantes presentan altos índices de dominio de estas técnicas de representación, ya que siguen adecuadamente en más de un 70%, los procedimientos y reglas establecidas para ellas. Por otra parte, como lo reportan más adelante Pérez Pérez y Monzalvo López (2010), “...*el uso de las diversas herramientas de diagramación, presenta una utilización de poco más del 61% en promedio, por parte de los estudiantes*”. Es en este punto donde surgen diversas interrogantes, entre las cuales una de las más inmediatas sería: ¿En qué medida y por qué razones, son utilizadas cada una de las herramientas por los estudiantes? El responder esta pregunta podrá dejar ver cual de las técnicas mencionadas es la que ofrece las mayores prestaciones para representar algoritmos diversos. Por tal motivo, el presente estudio llevará a cabo una exploración sobre el nivel de utilización de las técnicas de diagramación, por parte de los estudiantes.

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

Como lo han concluido los trabajos de Pérez Pérez y Monzalvo López (2010), para los estudiantes un algoritmo cíclico es más complicado de diseñar que un algoritmo lineal, debido a que este último sólo consiste en una secuencia de pasos o sentencias en forma encadenada; sin embargo, en un algoritmo iterativo hay que diseñar ciclos, los cuales implican desarrollar un grupo de sentencias ordenadas y vinculadas entre sí que comprenden una unidad en sí misma y que tienen un objetivo particular, lo cual no es una tarea fácil en muchos casos, ya que bucle se encuentra dentro de un programa, el cual tiene un objetivo más amplio.

En base a esta marcada diferencia, y con el propósito de medir el nivel de utilización de cada una de las herramientas de diagramación conocidas, se propone desarrollar un estudio de tipo cualitativo y cuantitativo, basado en el siguiente método:

- 1) Seleccionar dos programas tipo, uno lineal (secuencial-selectivo) y otro cíclico (secuencia-selectivo-iterativo)
- 2) En base a estos dos programas seleccionados, desarrollar un test exploratorio, que aborde aspectos relevantes sobre la utilidad de cada técnica de diagramación (diagrama de flujo, pseudocódigo y diagramas N-S)
- 3) Desarrollar un instrumento de evaluación para el test exploratorio, que tome aspectos cualitativos y los cuantifique (tabla de evaluación de ventajas y desventajas y tabla de registro de preferencias de uso de las herramientas), con la finalidad de recoger los datos plasmados por los estudiantes en los tests.
- 4) Aplicar el test a un grupo de estudiantes
- 5) Recabar los datos plasmados por los estudiantes en los test aplicados, por medio de los instrumentos de evaluación
- 6) Analizar los resultados obtenidos
- 7) Generar las conclusiones correspondientes.

## **Resultados experimentales**

La aplicación del test, se llevó a cabo con un grupo de 28 estudiantes de cuarto semestre, de la carrera de Sistemas Computacionales, perteneciente a la Universidad Autónoma del Estado de Hidalgo. Se seleccionaron estos estudiantes ya que se encuentran familiarizados con el uso de las técnicas de diagramación. Los resultados obtenidos son los siguientes:

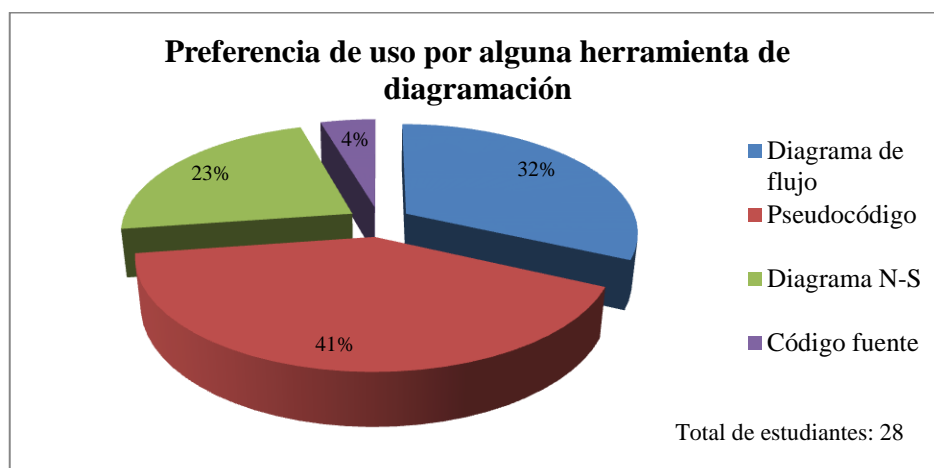
Según las opiniones de los estudiantes, se puede observar que la herramienta que más ventajas ofrece (ver tabla 1) es el diagrama de flujo con el 70.8% y el que menos es el diagrama N-S, apenas con el 55.7%, siendo una diferencia de poco más de 15%; lo inverso sucede en las desventajas: el diagrama N-S es el que presenta mayor desventajas (44.3%) y el diagrama de flujo las menores desventajas (29.2%).

Por su parte, el diagrama de flujo y el pseudocódigo presenta una diferencia de ventajas y desventajas de tan sólo poco más del 6%.

Herramienta	Ventajas (%)	Desventajas (%)
Diagrama de flujo	70.8	29.2
Pseudocódigo	64.7	35.3
Diagrama N-S	55.7	44.3

**Tabla 1. Comparación porcentual de las ventajas y desventajas de las técnicas de diagramación en estudio.**

En cuanto a las preferencias que manifestaron los estudiantes en cuanto a la utilidad que le atribuyen a cada herramienta de diagramación (ver figura 1), se tiene que la más preferida por los estudiantes es el pseudocódigo, seguido del diagrama de flujo y por último, el diagrama N-S; incluso hubo quienes expresaron su elección por el uso del propio código fuente, dejando de lado el uso de las técnicas de diagramación.



**Fig. 1. Resultados de la preferencia de uso de las herramientas de diagramación.**

## Conclusiones y trabajos futuros de investigación

Para concluir, sólo se dirá que los resultados confirman lo que menciona Joyanes Aguilar (1990) a este respecto: *“Debido a todas la ventajas que ofrecen las posteriores herramientas de diagramación, el diagrama de flujo ha ido decayendo en su empleo en favor del pseudocódigo y los diagramas N-S, y especialmente el pseudocódigo, que se ha hecho muy popular sobre todo desde la aparición de lenguajes de programación estructurados como Pascal, FORTRAN 77, True BASIC, Turbo BASIC, Quick BASIC o COBOL estructurado. Aunque el pseudocódigo no es tan claro como un dibujo, es una herramienta muy buena para el seguimiento de la lógica de un algoritmo y sobre todo muy fácil de convertir en programas escritos en lenguajes de programación específicos. Pero, pese a las ventajas innegables del pseudocódigo, desde el punto de vista didáctico, el diagrama de flujo*

y en menor medida el diagrama N-S, son una excelente herramienta para el aprendizaje de la programación” (Joyanes Aguilar, 1990).

Como lo menciona la investigación de Pérez Pérez, Fuentes Gálvez y Moreno Gutiérrez (2008), el conocimiento y aplicación adecuada de las técnicas de representación de algoritmos, no parece ayudar totalmente a resolver el problema que presentan los estudiantes, de su ineficiente capacidad para desarrollar algoritmos que reflejen correctamente las soluciones solicitadas. Hay que señalar que es necesario buscar mejoras o nuevas propuestas para las técnicas de representación de algoritmos, con el objetivo de reformar el proceso del desarrollo de algoritmos por computadora.

## Referencias

Forsythe, A.; Skeenan, T.; Organick, E.; Stenberg, W. (1975). “*Lenguajes de diagramas de flujo*”. Editorial LIMUSA. Segunda reimpresión. México. pp. 66.

Joyanes Aguilar, L. (1988). “*Metodología de la programación. Diagramas de flujo, algoritmos y programación estructurada*”. Primera edición. Editorial Mc Graw-Hill. México. p.14, 39.

Joyanes Aguilar, L. (1990). “*Problemas de la metodología de la programación*”. Primera edición. Editorial McGraw-Hill. México. p. 6, 7, 10, 21, 23.

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”. V Congreso Universitario en Tecnologías de Información y Comunicaciones 2010. CITIS. ICBI. UAEH. México.

Pérez Pérez, I.; Monzalvo López, C. A. (2010). “*Nivel de utilización de las técnicas de diagramación por parte de los estudiantes, en el diseño de algoritmos secuenciales, selectivos e iterativos*”. VI Congreso Universitario en Tecnologías de Información y Comunicaciones 2011. Área académica de Computación. ICBI-UAEH. México.