

“Análisis del desempeño de los estudiantes en el diseño y construcción de algoritmos secuenciales, selectivos e iterativos”

Área de Conocimiento: Computación Educativa

Isaías Pérez Pérez¹, Citlali Anahí Monzalvo López²

¹ 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, isaia spp7@hotmail.com

² 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: lex_any@hotmail.com

Resumen: *El presente trabajo exhibe la continuación del análisis de los resultados obtenidos en el estudio presentado por Pérez Pérez y Moreno Gutiérrez. (2009), el cual exploró la problemática presente en el diseño de los algoritmos secuenciales, selectivos e iterativos, concebidos dentro del paradigma de la programación estructurada, por parte de los estudiantes de las asignaturas de programación estructurada en la UAEH. Para dar seguimiento a esto, se diseñó, aplicó y se llevó a cabo el análisis de un instrumento de evaluación, que se centra en abordar problemas donde se desarrollan algoritmos con sentencias secuenciales, selectivas e iterativas, con el propósito de identificar en los estudiantes su desempeño en el diseño de estos tipos de algoritmos.*

Palabras clave: *Diseño de algoritmos, Programación Estructurada, secuenciales, alternativos, iterativos.*

Introducción

La Programación Estructurada, como lo menciona Joyanes (1990), se define como “*el conjunto de técnicas para desarrollar programas fáciles de escribir, verificar, leer (legibles para el usuario) y mantener (posibilidad de modificar)*”.

La Programación Estructurada se basa en un teorema fundamental llamado el teorema de Böhm y Jacopini (1966) o “el teorema de la estructura”, el cual establece que un programa puede ser escrito utilizando sólo tres tipos de estructuras de control: secuenciales, selectivas y repetitivas. Estas estructuras se les llama de control debido a que realmente controlan la ejecución del programa.

Según esta filosofía de programación, cada una de las tres estructuras de control poseen sólo un punto de entrada y un punto de salida, lo que hará que los programas estructurados sean más fáciles de leer y de depurar (o poner a punto) (Joyanes, 1990).

Estado del arte y problemática presentada

Diversos autores de programación (Scheid, 1984; Lozano, 1986; Joyanes Aguilar, 1993; Bores y Rosales, 1993), mencionan que la interrelación de las tres estructuras de control pueden generar tres diferentes tipos de programas estructurados:

- a) *Lineales o secuenciales*. Son aquellos en los que no existen instrucciones de bifurcación y por consiguiente las instrucciones se ejecutan en la misma secuencia en que han sido codificadas
- b) *Alternativos o selectivos*. Estos permiten la ejecución de diferentes operaciones, dependiendo de que se cumplan (o no) determinadas condiciones que se producen en los datos de las entradas o durante el proceso. Según la condición que se cumple se realiza una serie de instrucciones diferentes
- c) *Iterativos o cíclicos*. Son aquellos programas en los que un grupo de instrucciones se ejecuta un número determinado de veces (de modo repetitivo), hasta que se cumple una cierta condición que indica el fin de las condiciones que se repiten cíclicamente; esto se denomina bucle, lazo o ciclo. La estructura de un programa cíclico suele constar de los siguientes bloques o fases: a) entrada de datos e instrucciones previas; b) lazo o bucle (conjunto de instrucciones que se repiten y ejecutan un número determinado de veces); c) instrucciones finales o el resto del proceso; y d) salida de resultados (Joyanes Aguilar, 1993).

Finalmente, como lo menciona Scheid (1984), “*desarrollar un algoritmo significa combinar apropiadamente las tres estructuras de control*”.

La presente investigación continúa con el análisis del trabajo presentado por Pérez Pérez y Moreno Gutiérrez (2009), en donde se escogieron dos problemas tipo: el primero comprendía solamente sentencias secuenciales y de selección; y el segundo, era del tipo de los programas cíclicos. La razón de haber seleccionado estos dos tipos de problemas, es porque reflejan diferentes niveles de dificultad en el diseño de los algoritmos, ya que los algoritmos secuenciales y selectivos no requieren tener en cuenta elementos repetitivos, como es en el caso de los algoritmos cíclicos y particularmente en estos últimos se ha observado de manera empírica, que no son fáciles de desarrollar por parte de los estudiantes.

Cuando se plantean a los estudiantes, los enunciados de los dos tipos de problemas seleccionados, se les pide que lleven a cabo los tres pasos del diseño de algoritmos,

fases propias del Método de Resolución de Problemas por Computadora (Joyanes Aguilar, 1993):

- 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.

Metodología o técnica usada

La metodología que sigue el presente estudio (previamente planteada en la investigación de Pérez y Moreno, 2009), 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

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:

Aunque en el caso del diseño del algoritmo secuencial y selectivo, los estudiantes tuvieron un éxito promedio de 69.8%, se puede observar que en el desarrollo de los pasos que comprende la fase de diseño de algoritmos (ver figura 1), queda evidente que conforme se avanza desde el Diseño Descendente hasta la Representación de Algoritmos, el desempeño de los estudiantes va disminuyendo significativamente.

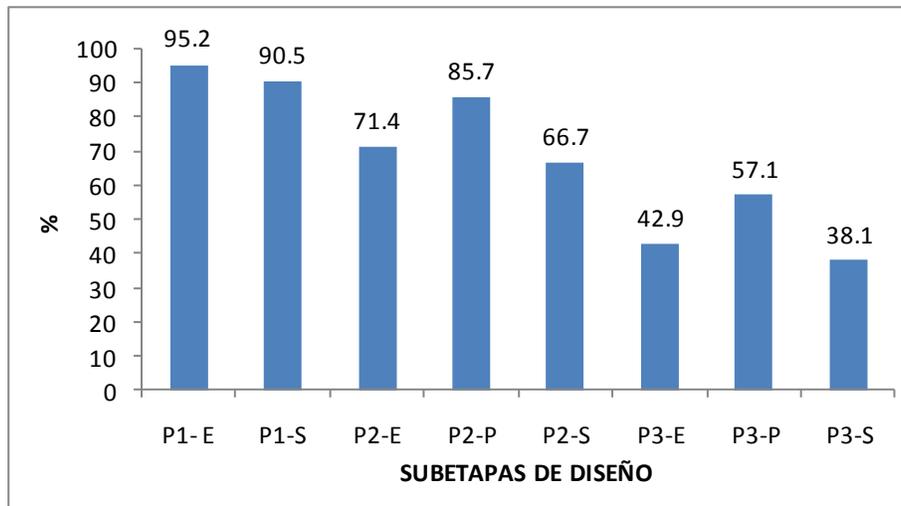


Fig. 1. Desempeño de los estudiantes en el problema de tipo secuencial y selectivo.

En la subetapa del Diseño Descendente, en donde se definen los subproblemas de manera verbal, la determinación de sus entradas y salidas, se logra alcanzar resultados de más del 90%, lo que hace pensar que los estudiantes tienen éxito en expresar de manera verbal, el diseño del algoritmo a construir.

En la subetapa del Refinamiento de los Subproblemas, en donde se trata de formalizar los planteamientos verbales de la subetapa anterior, los estudiantes muestran dos aspectos relevantes: aunque desciende su éxito en lograr el objetivo de esta parte, en poco más de 23% en promedio, el otro hecho es que pueden mejorar en poco más del 16%, el diseño del proceso que realizará el algoritmo, con respecto a la determinación de las entradas y salidas de este.

Finalmente, en la subetapa de Representación del Algoritmo, sucede algo similar a la fase de diseño anterior, ya que disminuye el desempeño de los estudiantes en aproximadamente 28% en promedio, pero a pesar de esto, mejora el diseño del proceso que realizará el algoritmo, es poco más del 16%, con respecto a la determinación de las entradas y salidas. Además en esta subetapa, el uso de las diversas herramientas de diagramación, presenta una utilización de poco más del 61% en promedio, por parte de los encuestados (ver figura 2).

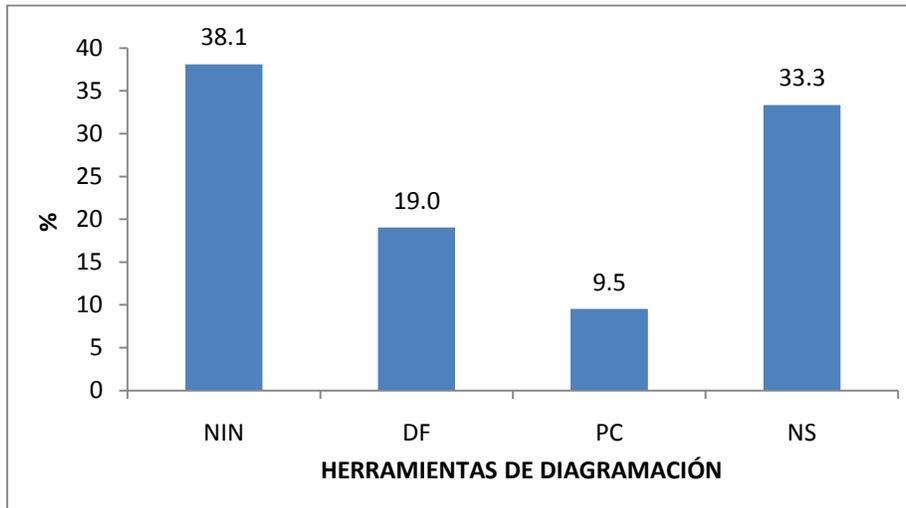


Fig. 2. Utilización de los estudiantes de las diversas herramientas de diagramación, en el problema secuencial y selectivo.

En el caso del diseño 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 algoritmo secuencial y selectivo (ver figura 3). El promedio de éxito en el desarrollo de los pasos para el diseño del algoritmo fue apenas del 21.7% en promedio, poco más de tres veces menos que el desempeño que tuvieron los encuestados, en el desarrollo del algoritmo secuencial y selectivo.

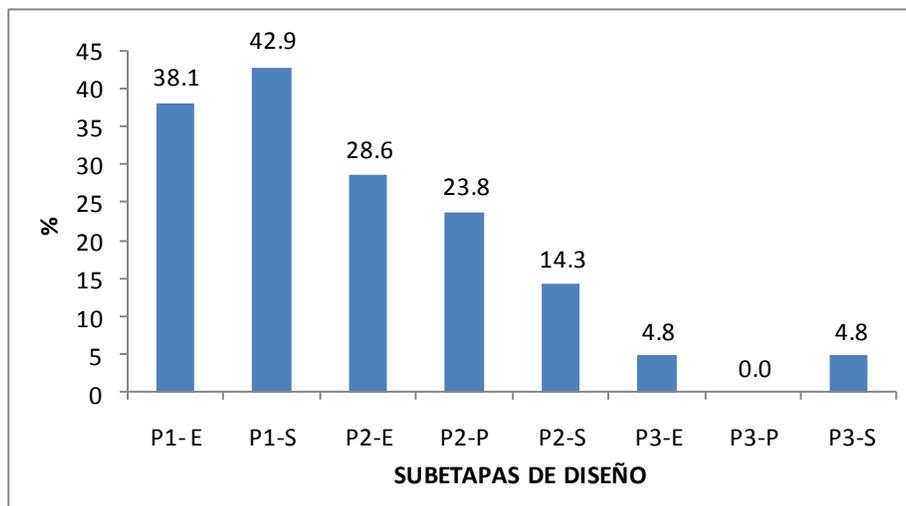


Fig. 3. Desempeño de los estudiantes en el problema de tipo cíclico.

Se puede observar que en la subetapa del Diseño Descendente, la determinación de las entradas y salidas, alcanza un éxito de poco más del 40% en promedio, menos de la mitad de lo que logran en el problema secuencial-selectivo (90%), lo que hace pensar que los estudiantes tienen desde un principio, dificultad en expresar de manera verbal el diseño del algoritmo a construir.

En la subetapa del Refinamiento de los Subproblemas, en donde se trata de formalizar los planteamientos verbales de la subetapa anterior, los estudiantes muestran una tendencia a la baja bastante marcada, en promedio del 22%, en comparación al desempeño que se tuvo en el problema secuencial-selectivo (74%) en esta misma subfase.

Finalmente, en la subetapa de Representación del Algoritmo, el desempeño de los estudiantes en el programa cíclico, disminuye hasta el 3% en promedio, siendo que en el problema secuencial-selectivo era del 46% en promedio, lo que indica un grave problema de cómo desarrollar el producto de esta subfase del diseño de algoritmos. Además, el uso de las diversas herramientas de diagramación en esta subetapa, es casi nula, ya que poco más del 90% de los encuestados no las utilizan (ver figura 4), indicando que las técnicas y herramientas de diagramación no parecen aportar elementos importantes para el diseño de algoritmos cíclicos.

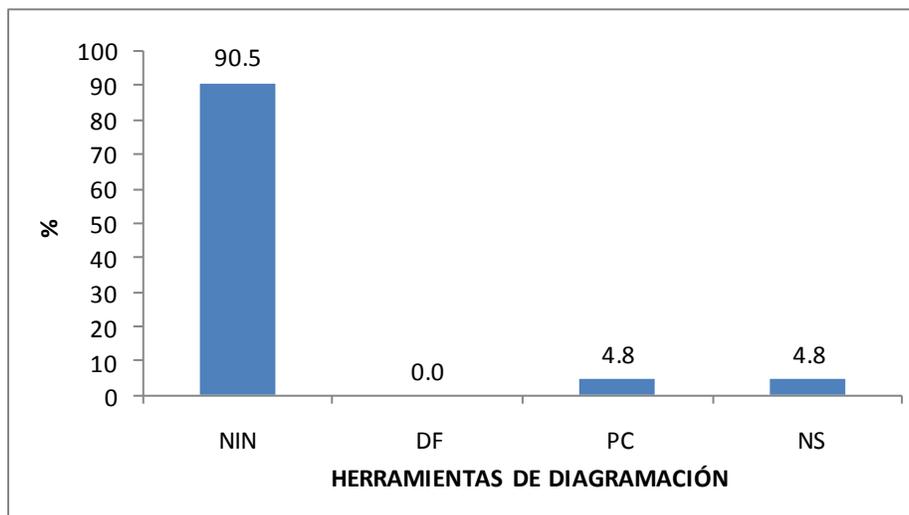


Fig. 4. Utilización de los estudiantes de las diversas herramientas de diagramación, en el problema cíclico.

Para terminar, se puede hacer una comparativa global del desempeño de los estudiantes en el diseño de los dos tipos de programas planteados en este estudio (ver figura 5). Por una parte, el diseño relativo al problema secuencial-selectivo (P-SS) muestra una tendencia de alto a medio, lo cual deja entrever que los estudiantes pueden concebir a

buen nivel, los algoritmos a construir de manera inicial, pero conforme se va avanzando en su especificación, esta es más difícil de que la logren. En cuanto al diseño de los problemas cíclicos, el problema es mucho más crítico, ya que el desempeño de los estudiantes se caracteriza por su bajo nivel de éxito, conforme se avanza en tratar de concebir y especificar las diversas subetapas del diseño de algoritmos.

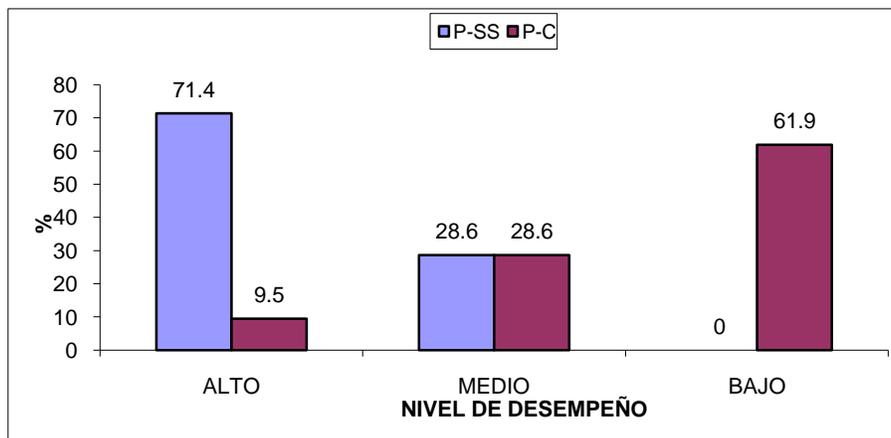


Fig. 5. Desempeño general de los estudiantes, en cada uno de los problemas tipo planteados.

Conclusiones y trabajos futuros

Para comenzar, los resultados obtenidos en la presente investigación refuerzan los ya obtenidos en la investigación previa de Pérez y Moreno (2009), en 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 secuencial, selectivo o cíclico.

Los resultados obtenidos, muestran que los algoritmos secuenciales-selectivos son relativamente fáciles de concebir por los estudiantes. Esto posiblemente se puede deber a que las personas piensan de manera muy similar a como estas sentencias se estructuran; no así en el caso de los problemas cíclicos, en donde los resultados son poco alentadores. El presente estudio hace evidente que en el uso de sentencias repetitivas es en donde los estudiantes tienen más problemas al tratar de implementarlas en el diseño de algoritmos. Esto es debido probablemente a que las personas no tienden a pensar en forma iterativa.

Para complementar la idea antes dicha, Scheid (1984), menciona al respecto: *“El hombre a menudo encuentra aburrido repetir, pero los computadores no comparten*

esta emoción. En efecto, es precisamente en esto donde sobresalen. Desarrollando algoritmos que hagan uso eficiente de la repetición, las máquinas se pueden programar para que hagan trabajos a gran escala". En este punto surge finalmente la pregunta: si es necesario programar a las computadoras con acciones repetitivas, pero las personas les es complicado concebir esta actividad de manera natural, ¿De qué manera deberían concebir los algoritmos cíclicos las personas, para que puedan plasmar claramente esa idea en los programas a desarrollar, los cuales más adelante serán alimentados a las computadoras para que los ejecuten?

Por otra parte, las técnicas y herramientas de diagramación clásicas para el diseño de algoritmos (diagramas de flujo, tablas N-S, pseudocódigo), parecen adecuarse adecuadamente a la forma de concebir los programas secuenciales y selectivos, en un poco más del 60% de los casos, pero parecen no aportar beneficios sustanciales en el diseño de programas repetitivos, ya que más del 90% de los estudiantes no las utilizaron para tal fin.

Debido a todo lo anterior, es necesario realizar más estudios sobre la problemática presente en el diseño de algoritmos cíclicos en investigaciones posteriores, ya que al parecer, los programas secuenciales y selectivos presentan un aspecto diferente en su concepción por las personas, en comparación con la forma de abordar programas iterativos o repetitivos.

Referencias

Bores, R.; Rosales, R. (1993). *"Computación. Metodología, lógica computacional y programación"*. Editorial McGraw Hill. México. Primera edición. pp.: 33 a 205. ISBN: 970-10-0225-3.

Joyanes Aguilar, L. (1990). *"Problemas de metodología de la programación"*. Editorial McGraw-Hill. Primera edición. México. pp.: 115 a 116. ISBN 84 - 7615 - 462 - 3.

Joyanes Aguilar, L. (1993). *"Fundamentos de programación. Algoritmos y estructuras de datos"*. Editorial McGraw-Hill. Primera edición. México. pp.: 76, 78,79, 81.

Lozano, R., L. (1992). *"Diagramación y programación estructurada y libre"*. Editorial McGraw-Hill. Tercera edición. México. pp.: 264.

Pérez Pérez, I., Moreno Gutiérrez, S. S. (2009). *"Estudio de la problemática relativa al uso de las sentencias secuenciales, selectivas e iterativas, en el diseño de algoritmos"*. IV Congreso Universitario en Tecnologías de Información y Comunicaciones 2009. Área académica de Computación. ICBI- UAEH. México. ISBN: 978-607-482-058-4.

Scheid, F. (1984). *"Introducción a la ciencia de las computadoras"*. Segunda Edición. Editorial McGraw Hill. México. pp.: 40 a 42. ISBN: 968-451-399-2.