

# “Estudio sobre la problemática en los enunciados de los problemas de programación”

Área de conocimiento: Programación de computadoras-Computación Educativa

Autores: M. en C. Isaías Pérez Pérez -ICBI -UAEH

Calle la atarjea s/n. Velasco. Mpio. de Omitlán de Juárez, Hgo. C.P. 43560

Tel. Cel. 044-771-114-8569

e-mail: [isaiaspp7@hotmail.com](mailto:isaiaspp7@hotmail.com)

M.C.C. Silvia Soledad Moreno Gutiérrez – Campus Tlahuelilpan -UAEH

L.I. América Fuentes Gálvez -ICBI –UAEH

Institución donde se realizó el trabajo:

Instituto de Ciencias Básicas e Ingeniería - Universidad Autónoma del Estado de Hidalgo

## Resumen

Es una situación conocida que la iniciación en las asignaturas de programación de computadoras suele ser casi siempre problemática (Joyanes Aguilar, 1991, p. XI). Esto se debe a que no se entienden de primera instancia, la definición y los elementos de los problemas que se pretenden solucionar (Lozano R., 1992, p. 13). En el presente trabajo, se presenta un estudio basado en la aplicación y análisis de un instrumento de evaluación, previamente diseñado, en estudiantes universitarios de la asignatura de programación, con el fin de identificar en que medida éstos poseen algunos de los elementos conceptuales iniciales y necesarios para el diseño de algoritmos de computadora.

## Palabras clave:

Programación, problemática, método, sistema.

## Introducción

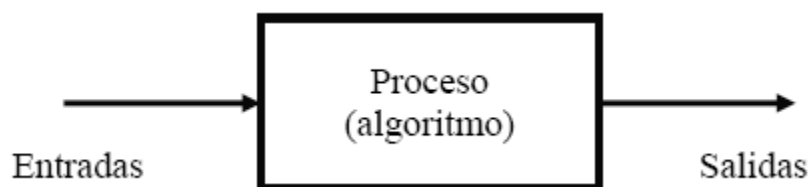
Es bien conocido que en el ambiente de la enseñanza de las asignaturas que abordan el diseño de algoritmos y programación en lenguajes de computadora, es un problema ampliamente conocido la alta dificultad de llevar adecuadamente la práctica de las técnicas de diseño de algoritmos, debido a los malos hábitos que los estudiantes de estos temas acostumbran seguir.

Para atacar esta problemática existente, las técnicas de la programación estructurada desarrolladas por Wirth, Dijkstra, Hoare, Bohm y Jacopini, entre otros, actualmente se recogen y aplican en lo que se conoce como el Método de Resolución de Problemas por Computadora (Joyanes Aguilar, 1993, p. XV).

## Estado del arte y problemática afrontada

La enseñanza de las asignaturas de programación, es un tema obligado en los cursos universitarios de hoy en día. Es de tan vital importancia su estudio, que reconocidos autores, han identificado el crucial papel que juegan estos temas en la educación de las ciencias computacionales; tal es el caso de Luis Joyanes Aguilar (1993), reconocido profesor y autor español, cuando menciona: *“Para el estudiante universitario, las asignaturas de programación se convierten en asignaturas clave en su formación, cuyo éxito o fracaso influirá decisivamente en el resto de sus estudios. La vital importancia de ésta asignatura obliga a un replanteamiento y estudio profundo en el que deberán intervenir esencialmente las modernas técnicas de programación y las características mas notables de los lenguajes de programación”* (Joyanes Aguilar, 1993, p. XV).

Las modernas técnicas de programación se ven hoy reflejadas primordialmente en el “Método de Resolución de Problemas por Computadora” (Joyanes Aguilar, 1993 y Watkins, 1984), el cual consta de 3 fases principales (ver fig. 1), visualizando a la construcción de algoritmos como la definición clásica de un sistema.



**Fig. 1. Fases de diseño y construcción de algoritmos, según el método de resolución de problemas por computadora**

Este método consta de las siguientes fases:

- 1ra. Fase: Análisis del problema
- 2da. Fase: Diseño de un algoritmo
- 3ra. Fase: Resolución del problema con computadora

La primera de ellas, el análisis del problema, trata sobre definir y acotar adecuadamente el problema a resolver, en correspondencia con una solución que se puede construir por computadora; a continuación se procede a definir las entradas y salidas que deberá tener el algoritmo a diseñar.

La segunda fase, el diseño del algoritmo, aborda la definición de los elementos particulares que deberá llevar el proceso del algoritmo; esto se realiza inicialmente como una jerarquización de los pasos del algoritmo a diseñar (diseño descendente); después se trata de representar de manera más detallada cada una de las instrucciones o pasos del algoritmo, de manera parecida al pseudo código. Finalmente, se utiliza alguna de las herramientas de diagramación de algoritmos, para hacer una representación grafica de el, con la finalidad de lograr un mejor entendimiento y una adecuada documentación del mismo.

La tercera fase, resolución del problema con computadora, trata sobre la codificación del algoritmo en un lenguaje de programación específico, su compilación y ejecución, así como

de la depuración de los errores que pudiese tener. Esta fase es la que construye un producto de software como tal.

El presente estudio pretende abordar de manera inicial, la primera etapa del método mencionado, dejando las dos restantes, no por falta de interés en la investigación de la problemática que presentan, sino con fines de acotar de alguna manera el estudio de la problemática de la etapa de análisis del problema. Entre los problemas que se presentan en esta primera fase, se puede destacar lo que menciona Letvin Lozano (1992), autor colombiano, al respecto: *“Los estudiantes creen que en llevar a cabo las técnicas de representación de algoritmos (diagramas de flujo, pseudo código, tablas N-S, etc.) radican todos sus problemas, pero en realidad el hecho es que no entienden correctamente los enunciados de los problemas a los que se enfrentan”* (Lozano R., 1992, p.13).

Tomando como hipótesis del estudio, lo mencionado por Lozano (1992), se procedió a llevar a cabo una investigación de campo, utilizando un test de evaluación escrito, basado en el método de resolución de problemas por computadora, en su primera fase (análisis del problema), el cual fue aplicado a estudiantes de las materias de programación, de las carreras de Sistemas Computacionales y de Ingeniería Industrial, impartidas en el Instituto de Ciencias Básicas, perteneciente a la Universidad Autónoma del Estado de Hidalgo, con el fin de ver en que grado los estudiantes comprenden el problema planteado, así como el número de fallas que tienen al determinar las entradas y salidas necesarias para el diseño de algoritmos.

### **Metodología o técnica usada**

La metodología del estudio es la siguiente:

- Selección de cuatro problemas tipo a plantear.
- Desarrollo del test de evaluación, presentando los cuatro problemas previamente escogidos
- Aplicación del test a grupos de estudiantes de las asignaturas de programación
- Recolección de los resultados obtenidos en los tests
- Análisis y conclusiones de los resultados

Los cuatro problemas tipo se numeraron en orden ascendente. Los problemas 1 y 2 presentaran un enfoque clásico de lo que se busca obtener de resultados (salidas) y los datos de los que se dispone para la construcción del algoritmo (entradas). Son aquellos problemas que aparecen comúnmente en los textos sobre el tema. Los problemas 3 y 4 presentaran un enfoque no ortodoxo a como lo presentan los textos; dan como entradas datos que comúnmente se definen como salidas y las salidas solicitadas serán datos que otros problemas proponen como entradas. En cada uno de los problemas presentados, se le presenta al estudiante una lista de opciones de entrada de datos y salida de resultados, siendo solo una la correcta, y las demás opciones parecidas a la correcta; estas incluyen datos que no aporta la definición del problema, aspectos que son propios del diseño del proceso del algoritmo o datos que no corresponden a la entrada o a la salida esperadas. Todo esto con la finalidad de observar en que medida los estudiantes tienen claro la definición de entradas y salidas de los posibles algoritmos a diseñar.

### Resultados experimentales

La población de estudio fue de 155 estudiantes de la asignatura de programación estructurada.

En cuanto a los problemas 1 y 2 (ver fig. 2), denominados de tipo clásico, se pudo observar que los estudiantes determinan el 2.6% más las salidas de resultados de los algoritmos presentados, que las entradas que estos requieren. Por el contrario, en los problemas 3 y 4 (ver fig. 3), denominados de tipo no clásico, los estudiantes determinan mejor las salidas en un 27.7%, que las entradas, lo cual hace suponer que independientemente del planteamiento del problema, los estudiantes son más capaces de determinar las salidas solicitadas. Por otro lado, aproximadamente el 1% de los estudiantes no pudieron determinar las entradas y salidas de los problemas 3 y 4 planteados.

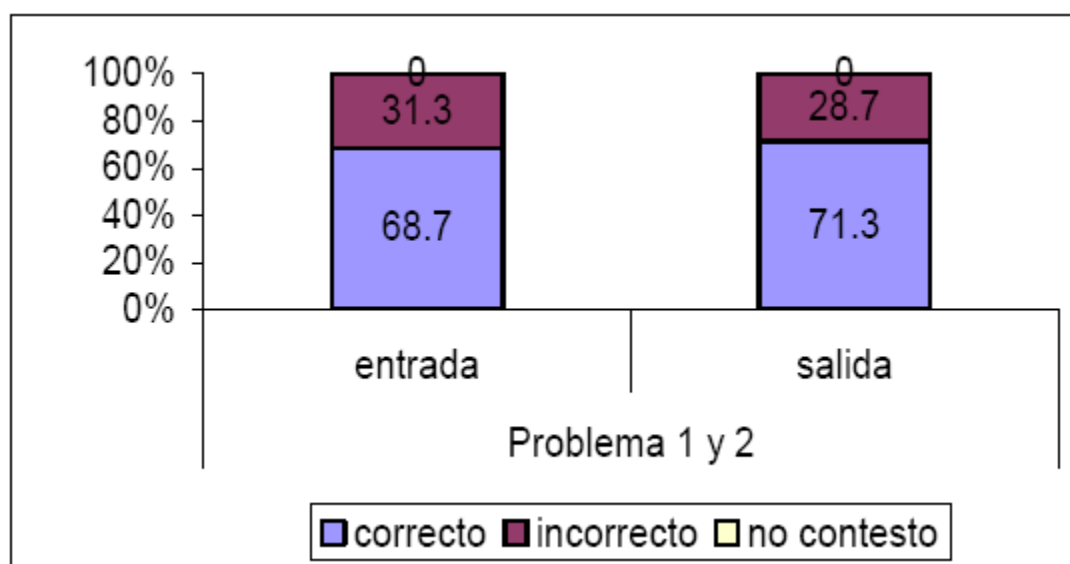


Fig. 2. Comparativa de entradas y salidas de los problemas 1 y 2 del test

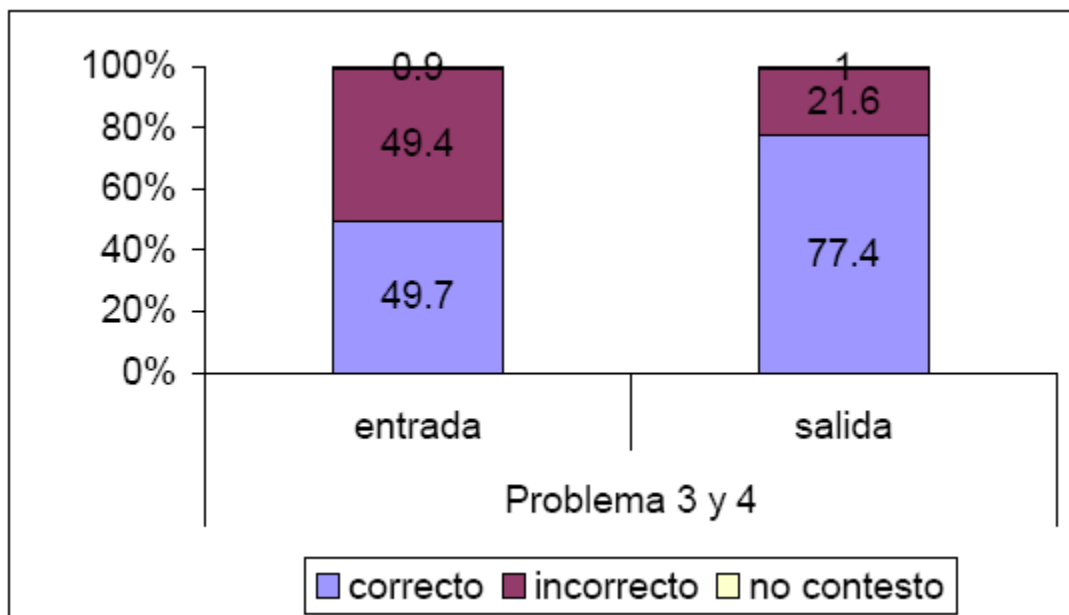


Fig. 3. Comparativa de entradas y salidas de los problemas 3 y 4 del test

Finalmente, al hacer una estimación global de los cuatro problemas presentados a los estudiantes (ver fig. 4), se puede observar que la determinación correcta de las salidas de los problemas planteados, es superior en un 15.1%, con respecto a la de las entradas, independientemente del estilo del planteamiento del problema. También se puede observar que solo un 0.5% de los estudiantes no pudieron determinar las entradas y salidas a identificar.

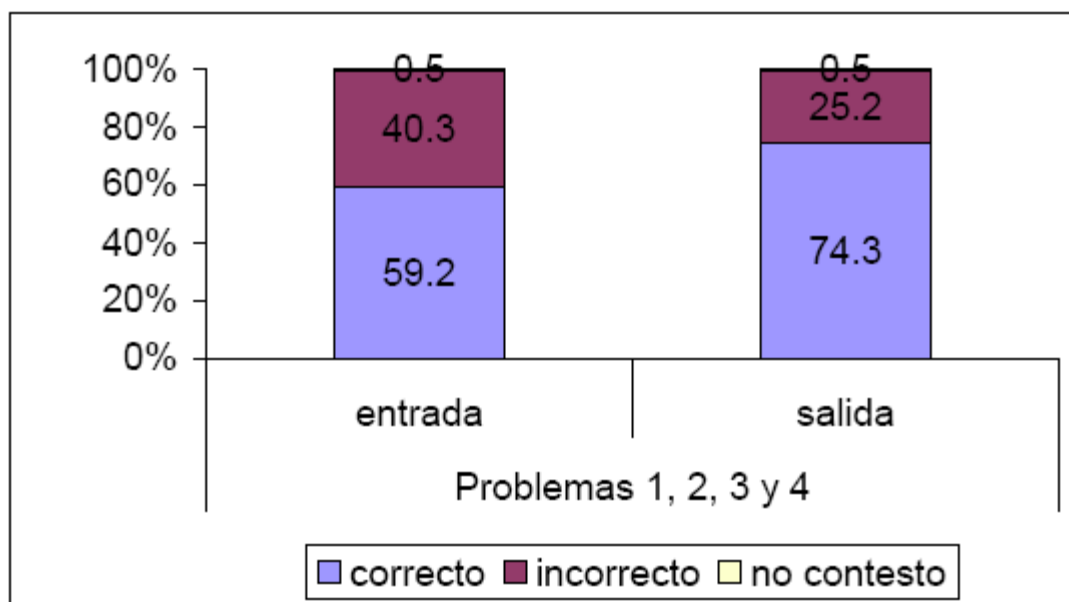


Fig. 4. Comparativa de entradas y salidas de los problemas 1, 2, 3 y 4 del test

## Conclusiones y trabajos futuros de investigación

Una primera conclusión que se puede apreciar, es que los estudiantes parecen haber aprendido “de memoria”, la etapa del “análisis del problema”, debido a que los problemas de tipo clásico, que son los más ampliamente enseñados en los textos y por los profesores, fueron los que más grado de aciertos tuvieron. En cambio, los problemas de tipo no clásico, mostraron una amplia diferencia de resultados, ya que al cambiar el planteamiento del problema, la determinación correcta de las entradas disminuyó en un 19%.

Por otra parte, los resultados obtenidos demuestran que independientemente del planteamiento de los problemas, en general, para los estudiantes les es más fácil determinar las salidas de resultados que las entradas de datos de los problemas planteados. Esto conduce a concluir que la aseveración de Letvin Lozano (1992), sólo es válida en parte, ya que los resultados obtenidos demuestran que aproximadamente el 60% de los estudiantes encuestados pueden determinar acertadamente las entradas de datos, el 75% pueden definir adecuadamente las salidas de resultados y solo el 0.5% no pudieron definir las entradas ni salidas, lo cual demuestra que la mayoría de los estudiantes entienden en buena medida, los enunciados de los problemas presentados, sea cual sea su planteamiento.

Las acciones, que se pretenden seguir en el desarrollo de la investigación, son:

- Hacer un estudio comparativo entre hombres y mujeres, con el fin de encontrar algún patrón específico.
- Determinar la influencia del planteamiento de los problemas, estudiando la relación entre las respuestas proporcionadas por una misma persona, en los cuatro diferentes problemas
- Entrevistar a los estudiantes que no presentaron fallas en el test, con el fin de obtener información relevante para el estudio
- Entrevistar y evaluar a los estudiantes que van más avanzados en los temas de programación, para encontrar algunos elementos conceptuales que sean relevantes para la investigación.

## Referencias

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

Joyanes Aguilar, L. (1991). “*Metodología de la programación. Diagramas de flujo, algoritmos y programación estructurada*”. Editorial Mc Graw-Hill. Primera edición. México.

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

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