

# ANÁLISIS DE ERRORES SISTEMÁTICOS DETECTADOS EN EL PROCESO DE APRENDIZAJE DE LA ALGORÍTMICA

Lic. A. Rosso, Ing. M. Daniele

(Asesoría pedagógica: Psp. Raquel Moressi)

Area de computación  
Facultad Ciencias Exactas Físico-Químicas y Naturales  
Universidad Nacional de Río Cuarto

## 1. Introducción

La enseñanza de la algorítmica nunca resultó una tarea fácil, y menos aún cuando no se usaba técnicas y metodologías que apuntaran a la correcta definición, interpretación y posterior planteo de la o las soluciones posibles para un problema dado.

Hasta hace poco tiempo, -y es una actualidad para algunos centros de educación- enseñar a programar consistía en mostrarle al educando un lenguaje de programación, en lo posible estructurado, con todas sus reglas sintácticas y semánticas, omitiendo o descuidando la enseñanza de métodos y técnicas que le permitiera a aquel reaccionar, interpretar y desarrollar soluciones a un problema planteado, independientemente del lenguaje que se utilizase.

De ésta manera, encontrar la o las soluciones a cada problema dado, requiere un ampísimo esfuerzo, por parte del programador: encontrar las líneas de código suficientes y correctas que arrojen alguna solución, pues está concentrado más en las reglas del lenguaje que en una verdadera y óptima interpretación y solución del problema.

Lo expuesto refleja, sintéticamente, los motivos que nos llevaron en el año 1993 a elaborar un proyecto pedagógico[1], para la asignatura Diagramación y Programación que se dicta en 1er. año de las carreras Ciencias de la computación, en el que la enseñanza de la programación se dirige a:

- Poner de relieve todos los factores que intervienen en la solución del problema, lo que implica entender el problema en todos sus detalles, para poder determinar los lineamientos principales que le darán solución.
- Inducir a utilizar técnicas que permitan la resolución de problemas planteados. Lo que implica conocer y saber aplicar distintas formas para plantear y resolver situaciones.
- Emplear esquemas correctos de tratamiento de secuencias, los que establecen una técnica en el uso de las estructuras de control.
- Simbolizar los procesos de solución en un lenguaje particular, parecido al lenguaje natural, que no es libre, sino que está sujeto a ciertas reglas de sintaxis y con una semántica. A este lenguaje lo llamamos "pseudo\_código". Este pseudo\_código estructurado, capaz de adaptarse a las especificaciones del problema, permitiendo escribir un algoritmo de fácil lectura a la vez que estructura la solución del problema y es independiente de un cierto lenguaje de programación
- Generar un algoritmo, expresado en el pseudo\_código, que establezca la íntima relación entre datos y las estructuras de control que permiten su tratamiento.

Como corolario de estos pasos, el algoritmo es traducido a un lenguaje de programación estructurado y es procesado en una computadora.

En el marco del desarrollo de este proyecto Pedagógico, que representa un avance respecto de propuestas de enseñanza anteriores, detectamos la presencia de algunos errores que aparecían sistemáticamente en grupos significativos de alumnos. Este problema, que constituye un paso más en los intentos por mejorar el proceso de enseñanza y de aprendizaje de esta asignatura, nos ha llevado a desarrollar el presente trabajo cuyo objetivo es:

- **Detectar y analizar las posibles causas de los errores sistemáticos que cometen los alumnos al estudiar la asignatura Diagramación y Programación**

## 2. Planteo del Problema

Con la metodología utilizada desde 1993, cuyo objetivo básico es lograr que los alumnos sean capaces de resolver problemas, planteando soluciones informatizables, y con la creencia de que el planteo de un problema supone una movilización y un desequilibrio para el sujeto, con la alegría y el sentimiento de superación ante la solución correcta y la construcción de un nuevo concepto sobre la base de los errores, cuando no se logra la respuesta (en un primer intento) y compartiendo con Azinian y otros [ ] que aprender

no es sólo encontrar procedimientos analíticos racionales sino también elegir estrategias, estilos de pensamiento y formas de representar la información es que hemos realizado un análisis de los errores más frecuentes que cometen los alumnos al estudiar Diagramación y Programación. Este estudio trata de evidenciar las causas y razones de tales errores proponiendo nuevos enfoques en el tratamiento de los temas conflictivos a fin de evitarlos y/o superarlos.

Nos proponemos detectar ya sea fallas en las estrategias seguidas y procedimientos utilizados o en una mala representación de la información.

Teniendo en cuenta las ideas de Dijkstra y Hoare, que sostienen que muchos errores de programación pueden evitarse haciendo a los programadores conscientes de métodos y técnicas que ellos hasta aquel momento aplicaban instintivamente y a menudo de forma inconsciente. Y convencidas de que formalizar los métodos y técnicas y recomendar su uso forjando una metodología de trabajo llevará a formular soluciones correctas es que nos abocamos al análisis de los errores.

### 3. Exposición de los errores sistemáticos producidos por los alumnos

En este apartado se expondrán los errores que sistemáticamente aparecen en las producciones de los alumnos. Previo a esto creemos conveniente dar a conocer al lector los pasos a tener en cuenta en la elaboración de una respuesta, dado que los errores suscitados tienen que ver con cada uno de estos.

Veamos cuáles son las dificultades con las que se enfrenta una persona que quiere solucionar un problema, planteando una respuesta informatizable.

Que debe hacer un programador cuando quiere resolver un problema?

- Debe realizar una descripción informal, pero precisa del problema a informatizar.
- El lenguaje informal utilizado para expresar el problema deberá ser reemplazado en etapas siguientes por un lenguaje artificial, formal y preciso, que cumpla con las reglas de la sintaxis y la semántica que le fueron impuestas. Debe describir el cálculo que permite obtener el resultado a partir de los datos.
- Afrontará la dificultad de modelizar su problema, o sea encontrar el modelo adecuado que le permita paso a paso ir poniendo progresivamente el problema en subproblemas que puedan ser computable. Este trabajo, el análisis, se va realizando en etapas, en la que va plasmando las decisiones que hay que tomar ante cada situación.

• Deberá ser capaz de asegurar la corrección de la solución. Las fuentes de error son múltiples, sólo a modo de ejemplo señalemos, errores en la lógica, errores de sintaxis, restricciones de presupuesto, costos etc. Es por ello la necesidad de usar técnicas que estén libres de errores

• Deberá aportar toda la documentación necesaria para el usuario

Para dar una buena respuesta a todas estas pautas es necesario proceder con un método, que facilite la tarea y le brinde seguridad en los resultados.

Una característica importante de los métodos o técnicas de programación es proveer al programador de una notación lo más próxima posible a los conceptos lógicos que utiliza una vez terminado el análisis del problema, notación muchas veces distante de los códigos que el ordenador utiliza.[3]

Las pautas enunciadas debieran ser formalizadas a fin de establecer una metodología de trabajo. La no observación de alguno de estos puntos con el debido cuidado, hace que se cometan errores en el planteo de la solución. Centrándonos ahora en los errores que aparecen en las producciones de los alumnos podemos inferir que los mismos tienen relación con el no cumplimiento de, por lo menos, alguno de los puntos señalados.

### 4. Clasificación de los errores detectados según los contenidos

A continuación mencionaremos los errores sistemáticos diferenciado los que aparecen independientemente del contenido tratado y los que son propios de alguno de ellos. El orden mostrado obedece a la ocurrencia de mayor a menor de los mismos.

#### • Errores comunes a todos los contenidos

- 1- Mezcla los tipos de variables, hace asignaciones y comparaciones inválidas
- 2- Cuenta mal. (En los problemas de conteo)
- 3- No puede expresar la solución con la sintaxis usada, falla la notación algorítmica especialmente en la expresión de las condiciones
- 4- No inicializa la/s variables que usa para presentar una condición, o plantea mal la condición
- 5- Faltan casos de tratar
- 6- Resuelve el problema para un determinado número de pasos, no puede generalizar
- 7- Ciclos sin fin, no avanza nunca dentro del ciclo
- 8- Confunde el uso de los operadores Y (and) y O (or), de > (mayor) y < (menor)
- 9- No inicializa variables utilizadas como contadores o acumuladores
- 10- Actualiza o lee la variable índice utilizada para un ciclo PARA

(Para <var> desde ii hasta is hacer)

- 11- No muestra los resultados obtenidos
- 12- Uso de variables no declaradas
- 13- No cierra los ciclos
- 14- Avanza en un ciclo sin considerar la condición que indica la marca de fin de la estructura utilizada
- 15- División por 0

• **Errores por contenidos específicos**

**Contenido I:** Lenguaje Natural. Estructura de Algoritmos. Manejo de lenguaje algorítmico propuesto. Composición secuencial, condicional, alternativa y selectiva. Funciones. Utilización de funciones en la construcción de algoritmos.

- 16- Confunde, mezcla el concepto de función con el de acción
- 17- Lee en el cuerpo de una función o acción las variables que pasa como parámetros
- 18- Define variables como parámetros de una función y en el cuerpo de la misma usa otras
- 19- Mezcla estructuras como SEGÚN <var> HACER con SI <cond> ENTONCES SINO
- 20- Define una función que no devuelve ningún valor

**Contenido II:** Composición de acciones. Composición iterativa. Tratamiento de secuencias. Máquina de caracteres. Arreglos unidimensionales y bidimensionales.

- 21- Mal manejo de los índices cuando se utiliza arreglos
- 22- Confunde filas con columnas

**Contenido III :** Archivos (Ficheros). Estructuras de datos, lista, pila, cola.

- 23- Mal enganche de los datos en listas encadenadas

**Fuentes de información**

Las fuentes de información que utilizamos para elaborar esta lista de errores posibles, son:

1. El análisis de una muestra de 250 evaluaciones escritas que realizan los alumnos (desde 1993), para cumplir con las exigencias de la asignatura Diagramación y Programación, que se dicta en el primer año de las carreras de ciencias de la computación de nuestra universidad.
2. Los aportes que realizan los docentes de la asignatura, según lo observado en las clases de aula y en las consultas individuales.
3. Las falencias detectadas cuando se evalúa a los alumnos en el examen final para la aprobación de la asignatura.

**5. Posibles razones o causas de los errores cometidos**

Las razones o causas de la ocurrencia de los errores que se desarrollan a continuación son inferencias que realizamos desde nuestra interpretación teórica del problema. Su verificación requerirá un trabajo posterior.

I) Las causas de éstos errores es que el alumno, que lo comete, no alcanzan a identificar los casos posibles para determinar la utilización de la estructura correcta. Si para una condición dada los casos posibles son dos, si la unión de estos casos dan el todo, deberá utilizar la composición alternativa ( Si <condición> entonces <consecuencia1> sino <consecuencia2>). Si independientemente de los casos posibles solo interesa un caso, deberá utilizar la composición condicional (Si <condición> entonces <consecuencia1>). Mientras que, si se requiere dar una respuesta a todos los casos posibles deberá usar la composición selectiva (Según <var> hacer ). Si el alumno no puede determinar todos los casos posibles para solución a un problema dado, tampoco podrá realizar un buen planteo de las condiciones.

(Justificación errores: 8, 19)

II) La falta de interiorización con el pseudo código utilizado provoca errores de sintaxis.

(Justificación errores: 3, 10)

III) El alumno tiene problemas con la definición formal de función y su diferencia con las acciones, ya que no tiene una clara conceptualización del concepto matemático de función. La dificultad para

establecer la diferencia entre parámetros formales y parámetros reales se produce, puesto que no establece la diferencia entre caso particular y caso general.

(Justificación errores: 20, 16, 17, 18)

IV) El sujeto trata de resolver el problema preocupado por los detalles de la sintaxis y no analiza los casos particulares atinentes al problema (mira al problema como un todo), tratando de encontrar un modelo que lo solucione.

El sujeto no puede realizar las combinaciones posibles abstrayéndose de los objetos mismos. [7]

(Justificación errores: 5)

V) Para representar una secuencia en memoria hay que establecer siempre la dirección del sucesor de un elemento dado. Esta dirección es lo que permite tener unida toda la secuencia en los tratamientos de inserción y supresión de elementos.

Cuando esta representación se hace por medio de punteros es necesaria una abstracción de la representación del espacio de memoria.

Esta manipulación de datos es relativamente simple gráficamente, pero no es tan sencilla cuando se lo codifica, se necesita guardar o memorizar las direcciones de los datos (actual, anterior, siguiente, etc). La no memorización correcta de éstas direcciones provoca la pérdida de una parte o de toda la secuencia.

La razón de cometer este error es no tener claramente identificada la técnica a utilizar para los movimientos habituales de las estructuras que los utilizan (inserción, supresión, listado, etc). No tienen internalizado el mecanismo a utilizar; aunque gráficamente pueden plantearlo la abstracción al código no se puede realizar, quedándose solo en la etapa de lo gráfico.

(Justificación errores: 23)

VI) El error se comete porque no usa, o no respeta, los esquemas de recorrido de secuencias. En realidad, el sujeto no puede modelizar su problema, ajustándolo a un modelo correcto y probado ya existente.

(Justificación errores: 7, 14, 15)

VII) Hay errores que provienen de una interpretación incorrecta de las consignas, quizás debido a una falta de hábito, por ser universitario de primer año a realizar esta tarea sin ayuda.

(Justificación errores: 11)

VIII) No puede analizar el proceso de abstracción. Para algunos casos no puede abstraerse de ejemplos particulares realizando una formalización o caracterización global.

(Justificación errores: 6)

IX) Cuando se trabaja con arreglos se maneja una estructura que identifica sus elementos con un índice. Sobre el conjunto de los índices está definida una relación de orden, que induce un orden en la ubicación de los elementos. Los errores en el manejo de estos valores puede ocasionar dos problemas:

1. Que se busquen elementos fuera del rango permitido.
2. Mala manipulación de la información

Estos errores muestran que el alumno no incorpora el concepto y lo relaciona con un "ordenamiento en la ubicación de los datos".

(Justificación errores: 21, 22)

X) Cuando uno define una variable debe definir de que tipo es; con ello se caracteriza el conjunto de valores al que pertenece. El tipo de un dato no puede ser deducido del contexto, cuando se programa, entonces debe declararse explícitamente.

Los alumnos que no tienen claro lo expuesto en el párrafo anterior, mezclan los tipos en asignaciones o comparaciones ya que no pueden detectar claramente el conjunto de valores de los cuales las variables toman sus valores (iniciales o finales)

(Justificación errores: 1, 4, 9)

## **CONCLUSIONES FINALES**

Desde el año 1993 se desarrolla en la asignatura Diagramación y Programación de las carreras de Ciencias de la Computación de nuestra Universidad, un proyecto pedagógico que contempla una abordaje de la enseñanza de los contenidos superadora de otra modalidad que consideramos no suficientemente adecuada para la enseñanza de los temas.

En el marco del desarrollo de dicho proyecto es que aparecieron errores sistemáticos en las producciones de los alumnos, que analizamos en este trabajo a los fines de acrecentar el mejoramiento de la enseñanza y el aprendizaje. Estos errores son de distinta naturaleza: tienen que ver tanto con las estrategias seguidas y procedimientos utilizados, como con una representación inadecuada de la información ; y están vinculados con todos los contenidos o con algunos en particular.

En base al análisis realizado, surge claramente la necesidad de continuar replanteándonos algunos aspectos de nuestra metodología de enseñanza en los que operen transformaciones al conocimiento que estamos enseñando, que posibiliten la superación de los errores mencionados en la medida en que éstos no estén directamente vinculados a otras razones que no sean de naturaleza didáctica.

## Bibliografía

- [1]- A ROSSO, M. DANIELE, G. ROJO, G. FRASCHETTI. Proyecto Pedagógico Innovador para la enseñanza de la Algorítmica. Fac. Cs. Extas. Fco-Qcas y Naturales.Dpto. Matemática U.N.R.C. 1993
- [2] A. ROSSO, J. GUAZZONE. Errores que se cometen al resolver un problema con estructuras de repetición. Dpto. de Matemática. Fac. Cs. Extas. Fco-Qcas y Naturales. U.N.R.C. 1995
- [3]- P.C. SCHOLL - J.P. PEYRIN. Esquemas Algorítmicos Fundamentales. Secuencias e Iteración. Ed. Masson, 1991.
- [4]- J.D. NOVAK - D.B. GOWIN. Learning how to learn. New York. Cambridge University Press, 1984
- [5]- D.P. AUSUBEL - J.D. NOVAK - H. HANESIAN. Educational Psychology: A Cognitive View. 2nd. Ed. New York. Holt , Rinerhart anf Winston. 1978.
- [6]- Ingenierío de l U.N.Comahue.
- [7]- B. INHELDER Y J. PIAGET. de la lógica del niño a la lógica del adolescente. Ed. Paidos, 1972.
- [8]- MARK ALLEN WEISS. Estructuras de datos y algoritmos. Ed. Addison-Wesley Iberoamericana, 1995.
- [9]- J. GLENN BROOKSHEAR. Introducción a las Ciencias de la Computación. Ed. Addison-Wesley Iberoamericana, 1995.
- [10]- NIKLAUS WIRTH. Algoritmos + Estructuras de datos = Programas. Ed. del Castillo S.A., 1980.
- [11]- CAIRÓ / GUARDATI. Estructuras de Datos. Ed. McGraw-Hill Interamericana de México, 1993.
- [12]- AZINIAN-BRENTA-ALVAREZ. Tecnología Informática en la escuela. Ed. AZ-Editora. 1995.