

# Estrategias para la enseñanza de la programación

Norma Moroni – Perla Señas  
[nem/ips]@cs.uns.edu.ar

Laboratorio de Investigación y Desarrollo en Informática y Educación (LIDInE)  
Instituto de Investigación en Ciencias y Tecnología Informática (IICTI)  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Bahía Blanca

## Resumen

La complejidad de los programas que se desarrollan actualmente produce la necesidad de iniciar a los alumnos en un camino que los conduzca a utilizar efectivas técnicas de programación. Es importante para ello poner énfasis en el diseño previo. Como se ha comprobado, una estrategia valedera es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema. Esto genera una primera etapa de la programación que resulta un tanto tediosa para los alumnos que están ávidos de utilizar la computadora. Si bien no aparecen dificultades graves con el aprendizaje de esta técnica, se puede comprobar que no resulta una tarea trivial obtener un algoritmo semánticamente correcto. El hecho de reescribir los algoritmos hasta ponerlos a punto es operativamente complicado cuando se trabaja con lápiz y papel. Además, comprobar la corrección del algoritmo presenta inconvenientes importantes. Es difícil, mental o gráficamente, representar las acciones del algoritmo en ejecución de manera totalmente objetiva, sin dejarse llevar por la subjetividad, fundamentalmente cuando el que lo hace es el propio autor del algoritmo. Por otra parte, se ha comprobado que el uso del método global para el aprendizaje del lenguaje de programación, ahorra tiempo y esfuerzo. Con el propósito de trabajar especialmente sobre los aspectos mencionados se creó un Ambiente de Aprendizaje con un editor interactivo de algoritmos, un constructor automático de trazas y un traductor de algoritmos a programas en lenguaje Pascal. Se presentan en este trabajo, los resultados obtenidos en una experiencia de campo diseñada para comprobar la efectividad de la aplicación del entorno de programación mencionado.

## 1. Introducción

La complejidad de los programas que se desarrollan actualmente produce la necesidad de iniciar a los alumnos en un camino que los conduzca a utilizar efectivas técnicas de programación. Es importante para ello poner énfasis en el diseño previo.

Una estrategia valedera es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema [Levy].

Esto genera la inclusión de una etapa previa a la programación que resulta un tanto tediosa tanto para los alumnos ávidos de utilizar la computadora como para aquellos que la utilizan habitualmente. Además, si bien no aparecen dificultades graves con el aprendizaje de esta técnica [Rueda], no resulta una tarea trivial obtener un algoritmo semánticamente correcto, sino que para lograrlo se requieren sucesivos refinamientos. El hecho de reescribir los algoritmos hasta ponerlos a punto es operativamente complicada cuando se trabaja con los elementos tradicionales como lápiz y papel, tiza y pizarrón, etc. No obstante, lo que presenta un inconveniente importante es comprobar la correctitud del algoritmo. Es difícil, mental o gráficamente, llevar a cabo las acciones del algoritmo en ejecución de manera totalmente objetiva sin dejarse llevar por la subjetividad de su especificación, es decir, tratando de olvidar el pensamiento que llevó al desarrollo y concentrándose exclusivamente en lo que se encuentra escrito.

Con el fin de minimizar las etapas que no aportan demasiado interés en lo que se refiere a la programación en sí, es decir, lo que corresponde concretamente a la resolución del problema, al diseño y a la formulación del programa, es importante poder contar con un editor de algoritmos que ayude al alumno en la especificación del mismo y que, además, permita comprobar su correctitud tanto como la detección de errores mediante la confección automática de trazas. El uso del editor interactivo de algoritmos [Moroni] presenta ventajas comparativas frente a la forma de especificación tradicional ya que favorece la escritura del mismo, contando para ello con plantillas de las distintas estructuras de control, datos de entrada, datos de salida, comprobación de tipos de datos, etc. Cuenta con un corrector sintáctico y semántico que asegura la escritura correcta del mismo y favorece al alumno dejando que centre su atención en la lógica del procedimiento; realiza automáticamente la traza de su ejecución a partir de los datos de entrada que se le ofrezcan, y traduce el algoritmo a un programa en un lenguaje de programación, ayudando a reconocer sus construcciones y permitiendo mapearlas con las porciones correspondientes en el algoritmo. El aprendizaje del lenguaje de programación siguiendo el método global ahorra el tiempo y el esfuerzo que se aplica en la primera etapa de presentación, quedando sólo la puntualización de aquellos detalles que son más específicos y concretos.

Luego de completarse la construcción del mencionado Ambiente de Programación, se quiso comprobar su efectividad. Para ello, se realizó una experiencia comparada en cursos de carreras no informáticas, donde la disparidad de conocimiento y de motivación era importante. Se cotejaron dos cursos similares, uno de ellos funcionó como testigo empleándose en él una metodología tradicional, mientras que el otro se desarrolló en el Ambiente de Programación expuesto

## 2. Experiencia en el aula con el Editor de Algoritmos

Las estrategias de aprendizaje son procedimientos que presentan rasgos propios tales como:

- su aplicación no es automática sino controlada
- se componen de otros elementos más simples, que constituyen técnicas o destrezas. El uso eficaz de una estrategia dependerá en buena medida del dominio de las técnicas que la componen
- el uso selectivo de los propios recursos y capacidades disponibles. El alumno, al poner en marcha una estrategia, debe disponer de recursos alternativos, entre los cuales decide utilizar, en función de las demandas de la tarea de aprendizaje que se le presenta, aquellos que considera mejores. Sin una variedad de recursos no es posible actuar estratégicamente.

A partir del Editor de Algoritmos y su entorno de trabajo surge en forma natural el interrogante de saber si realmente existen ventajas comparativas entre el uso del editor y la confección de los algoritmos en forma tradicional. La respuesta a este interrogante se comenzó a construir a partir de las conclusiones obtenidas luego del desarrollo de una experiencia educativa de campo en la que se contemplaron y compararon ambas estrategias. Se realizó una experiencia en dos cursos iniciales de Programación de carreras no informáticas, donde la disparidad motivacional y de conocimiento es importante. En uno de los cursos se utilizó el Ambiente de Programación, y se comparó con otro curso de similares características donde no se lo usó.

Se consideraron tres tendencias de evaluación diferenciadas: diagnóstica, formativa y sumativa, a través de la aplicación de procedimientos cuali y cuantitativos. Esto permitió obtener información sobre el estado de los alumnos al iniciar la experiencia, sobre su progreso durante la implementación de la misma, y sobre la valoración de los logros alcanzados al finalizar el proceso.

### 2.1. Descripción

Se tomaron dos cursos de características similares y bajo la responsabilidad del mismo docente; en uno de los cursos se trabajó con el Ambiente de Programación descripto y en el otro de una manera tradicional. La experiencia se desarrolló totalmente en la misma institución educativa.

Los condicionantes de la tarea se mantuvieron constantes a lo largo del desarrollo de la experiencia, tanto la composición de los grupos con sus respectivos docentes como las características etarias (primer año de la carrera de Profesorado en Matemática) y la disponibilidad de recursos escolares (físicos y materiales).

Con el propósito de analizar el estado inicial de los alumnos, de acuerdo a las variables evaluativas explicitadas, se instrumentó una encuesta de diagnóstico, indagando acerca de sus conocimientos previos en la técnica de programación, uso de la computadora y su aplicación. A partir de la información obtenida en esta evaluación, el grupo de investigación decidió instrumentar dos sesiones de clases en las que se enfatizaba la estrategia de resolución de problemas, para lograr la construcción de una base común, en lo relativo a la comprensión de la técnica y a sus posibilidades de aplicación.

Durante el transcurso de la experiencia el grupo de investigación implementó una evaluación de tipo formativa, realizando observaciones de diversas instancias áulicas, en las cuales se consideró: interés de los alumnos, clima del aula, impacto de la novedad en el uso del editor, actitud de los docentes, grado de correlación entre la comprensión de la técnica y su aplicación.

En la evaluación sumativa, en la cual se trabajó con los productos elaborados por los alumnos (algoritmos tradicionales y algoritmos editados) se tuvieron en cuenta: el modelo de resolución, la corrección de las relaciones establecidas, la correctitud del algoritmo, el tiempo de elaboración y el proceso de refinamiento.

## **2.2. Evaluación de la Experiencia**

En la evaluación de proceso se observó que los alumnos del curso que trabajó en el Ambiente de Aprendizaje demoraron un poco más de tiempo en la elaboración de los algoritmos iniciales ya que debieron interiorizarse sobre aspectos de la plataforma. Superado este período, los estudiantes involucrados en este curso encontraron mayoritariamente poca dificultad en la tarea de confección de los algoritmos, superaron las diferencias de conocimientos existentes y se adaptaron a la estrategia de programación con algoritmos, mientras que los alumnos del curso tradicional tuvieron dificultades en la construcción y refinamiento de los mismos; fundamentalmente presentaron resistencia a seguir usando la estrategia de algoritmos una vez que se introdujo en el curso la etapa de implementación. Si bien resultó más lento el comienzo en el primer curso que en el segundo, el tiempo se recuperó con creces en el momento de codificar los algoritmos en un Lenguaje de Programación. Aquí, los alumnos del primer curso se vieron aventajados, primero, porque sus algoritmos se codificaban automáticamente, y segundo porque tuvieron, desde el principio, la posibilidad de aprender paulatina e incrementalmente el lenguaje de implementación. Los alumnos del curso tradicional se vieron forzados a conocer gran parte del mismo antes de codificar sus algoritmos.

A pesar de que el período de adaptación fue más largo en la etapa de la algoritmia, el grupo que usó el editor demostró que el producto final era mucho más ordenado, prolijo, con facilidad para el refinamiento progresivo y la corrección. Además, se observó que la mayor diferencia a favor del grupo de edición se observó en la correctitud del algoritmo ya que este grupo tenía la posibilidad de verificar sus algoritmos por medio de la aplicación del ejecutor automático de trazas, mientras que el segundo grupo debía hacerla manualmente. Además, se observó que la aplicación del método global propende al uso continuo de la plataforma por parte de los estudiantes y desarrolla en ellos buenas aptitudes para la lectura y comprensión de programas.

La evaluación sumativa arrojó los resultados que se presentan en la siguiente tabla. Los valores se expresan en porcentaje sobre la población del curso:

PRUEBA	TÓPICO	GRUPO 1 Método tradicional			GRUPO 2 Con Ambiente de Aprendizaje		
		Rendimiento			Rendimiento		
		Bajo	Medio	Alto	Bajo	Medio	Alto
Diagnóstico	Manejo PC	14	76	10	10	79	11
	Motivación	42	41	38	38	44	18
	Conocimientos previos	87	13	91	91	6	3
Algoritmos Nivel 1	Correctitud	40	43	64	64	35	1
	Claridad	46	39	66	66	33	1
Algoritmos Nivel 2	Correctitud	38	45	39	39	48	13
	Claridad	52	32	31	31	53	16
Programas en Pascal	Correctitud	42	40	27	27	55	18
	Claridad	40	36	23	23	56	21

### 3. Conclusiones

Frente a la necesidad de incorporar nuevas estrategias para el desarrollo de programas cada vez más complejos, como lo es el empleo de algoritmos, es importante el uso de herramientas computacionales que ayuden a tal fin.

La experiencia realizada permite realizar las siguientes conclusiones:

- El uso del editor de algoritmos desde el principio: disminuye la ansiedad por el uso de la computadora tanto de los alumnos con conocimientos en computación como de los que no los tienen.
- Facilita la nivelación entre aquellos alumnos con conocimientos en computadoras y aquellos que no tuvieron acceso a ella.
- Fomenta el posterior uso de la estrategia de algoritmos aún después de aprender a desarrollar programas.
- Si bien se observa, en principio, mayor lentitud en el desarrollo de los algoritmos con la nueva propuesta, debido a la necesidad de aprender a usar la herramienta, el tiempo empleado se recupera en la etapa de prueba y programación.
- Fomenta la realización del chequeo del algoritmo, como etapa previa a la codificación del mismo

### 4. Bibliografía

[Aho] Aho, Alfred; Sethi, Ravi and Ullman, Jeffrey. Compilers: Principles, Techniques and Tools. Addison-Wesley. 1986.

[Apple] Apple, A. Modern compiler implementation in Java. Cambridge University Press. (1998).

- [Borland] Borland International Inc. Object Windows Programming Guide. 1992
- [Chavey] Darah Chavey. Beloit College. A Structured Laboratory Component for the Introductory Programming Course. SIGCSE BULLETIN ACM. 1991.
- [Díaz] Díaz Barriga Arceo, F. y Hernández Rojas, G. “Estrategias docentes para un aprendizaje significativo”. Mc. Graw Hill, Mexico. 2001.
- [Geitz] Robert Geitz. Concepts in the classroom, programming in the lab. ACM SIGCSE BULLETIN. 1994.
- [Joyanes] Joyanes, L.; Rodriguez, L. y Fernández, M.: Fundamentos de Programación. Ed. McGraw-Hill. 2002.
- [Lajoie] Lajoie, Susanne. “Computer Environments as Cognitive Tools for Enhancing Learning”. 1993. McGill University.
- [Levy] Lisa Levy Kortright. From Specific Problem Instances to Algorithms in the Introductory Course. SIGCSE BULLETIN ACM. 1994.
- [Medina ] Medina Duarte Roseclea, Tarouco Liane M. R., Amoretti Suzana Laboratório Virtual ASTERIX – resultados decorrentes da sua utilização como ferramenta cognitiva. X Congreso Argentino de Ciencias de la Computación. Buenos Aires 2004
- [Mitchell] Mitchell, John and Apt, Krzysztof. Concepts in Programming Language. Stanford University. 2004
- [Moroni] Moroni, Norma y Señas, Perla. Un entorno para el aprendizaje de la programación. CACIC. 1996.
- [Pittman] Pittman, Thomas; Peters, James; The Art of Compiler Design, Theory and Practice. University of Arkansas. Prentice-Hall International. 1992.
- [Terry] Terry, P. ‘Umbriel - Imperative Programming for unsophisticated students’. ACM SIGCSE BULLETIN. 1995.
- [Wu] Wu C. Thomas. Teaching OOP to beginners. Journal of Object-Oriented Programming. 1993.
- [Rueda] Rueda, S.; Castro, S y Zanconi, M. Resolución de Problemas y Algoritmos: notas de curso. 1994.