

**CACIC 2004 CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN
XII Ateneo de Profesores Universitarios de Computación**

TÍTULO:

**El desafío de favorecer los aprendizajes de los alumnos en
conceptos básicos de algorítmica y programación.**

Autores

Lic. Prof. Ariel Ferreira Szpiniak
aferreira@exa.unrc.edu.ar

Ing. Guillermo A. Rojo
grojo@exa.unrc.edu.ar

Departamento de Computación
Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Universidad Nacional de Río Cuarto
Ruta 36 Km. 601. Agencia Postal 3, Río Cuarto, Córdoba. C.P. 5800
Tel. 0358 – 4676529

Palabras Clave: Algorítmica, Programación, Aprendizaje, Enseñanza

Resumen

En este trabajo se presenta la propuesta que actualmente está llevando a cabo el equipo docente de la materia Introducción a la Algorítmica de las carreras de Analista en Computación y Licenciatura y Profesorado en Ciencias de la Computación que se dictan en la Universidad Nacional de Río Cuarto. La propuesta pretende favorecer los aprendizajes de los alumnos en conceptos básicos de algorítmica y programación puesto que se han detectado muchas falencias en la apropiación de tales conceptos cuando los alumnos deben aplicarlos en materias de años superiores.

La propuesta aquí presentada forma parte del proyecto de investigación e innovación de la enseñanza de grado aprobado por la Secretaría Académica y de Ciencias y Técnica de la Universidad Nacional de Río Cuarto. Las innovaciones a que se hace referencia son al interior de la asignatura y con relación al proceso de enseñanza y de aprendizaje que se ha desarrollado en años anteriores. Se exponen cuales son los lineamientos a seguir en el curso 2004 y cuales son los supuestos que sustentan la metodología a llevar adelante.

Problema que origina la innovación

En primer término reconocemos una evolución en la concepción y enseñanza en el desarrollo de programas de computación. En una primer etapa la enseñanza se basó en enseñar las particularidades del lenguaje que se utilizaba para programar. Posteriormente con la aparición de los lenguajes estructurados se impuso la programación estructurada. Aquí se vislumbraron dos tendencias que perduran hasta hoy en día: la primera planteó la enseñanza de la programación estructurada vinculada a un lenguaje estructurado en particular, la segunda se basó en una visión más amplia del problema de la construcción de programas y planteó la necesidad de independizarse del lenguaje de programación, definiendo un estándar llamado “pseudo-lenguaje”, con un grado de generalidad que permite en un segundo paso la traducción de la solución algorítmica desarrollada a diferentes lenguajes. Esta segunda tendencia permitió independizar la enseñanza de la programación de la enseñanza de los lenguajes de programación y fue la tendencia adoptada desde los inicios en nuestras carreras con el objetivo de lograr que los estudiantes fueran capaces de encontrar soluciones informáticas estructuradas.

Así mismo dentro de esta línea se hizo hincapié en la idea de utilizar máquinas abstractas y esquemas para el tratamiento de secuencias a los fines de lograr algoritmos correctos. En otra etapa de cambio se incorporó un mayor grado de formalismo con el fin de desarrollar algoritmos correctos o cuya correctitud pudiera ser demostrada aunque no de una manera rigurosa. Se introdujo en el diseño diversas técnicas que garantizaba al menos parcialmente esta correctitud o al menos permitían su verificación, aunque de una manera semi formal.

Con el ánimo de propender al desarrollo de algoritmos correctos se introdujeron con mayor énfasis los conceptos de estado inicial o precondition del problema, la que se debe cumplir para garantizar la terminación del algoritmo en un estado final o postcondición la cuál también es previamente definida y que expresa la solución que se pretende alcanzar. Asimismo y acompañando estos conceptos se introdujo el uso de invariantes para derivar la distintas partes que conforman a un ciclo. Para abordar ésta forma de resolución de algoritmos fue necesario introducir el uso de predicados y la aplicación de reglas de lógica.

Los esfuerzos por aproximar soluciones correctas tuvieron un efecto negativo, el énfasis puesto en este tipo de metodologías imposibilitó el trabajo con problemas complejos, le restó tiempo de desarrollo a los proyectos (desarrollo de programas más complejos y grandes) y a temas como tipos, acciones, funciones, pasaje de parámetros, arreglos, matrices, archivos, punteros y recursividad. Como consecuencia de ello y a pesar de la constante preocupación por la enseñanza de la programación se verificaron diversos indicadores que pusieron en evidencia una situación problemática que era necesario abordar, algunos de estos indicadores son:

- Alta deserción en los alumnos que inician la asignatura.
- Alta cantidad de alumnos que dejan la asignatura sin agotar todas las instancias evaluativas.
- Los alumnos que alcanzan la condición de regulares en la asignatura ingresan a segundo año con graves falencias en cuanto a la construcción de algoritmos y a conceptos fundamentales que luego son de utilidad a lo largo de toda la carrera y a su actividad profesional.
- Escasa destreza en los alumnos para desarrollar algoritmos de mediana o alta complejidad.
- Poca ejercitación a lo largo del curso como resultado del método aplicado para el desarrollo de algoritmos.
- Aumento en la complejidad de la resolución, originada en el propio método y que deviene entre otros factores en la dificultad para especificar estados intermedios en los algoritmos,

con un mayor énfasis en el caso de determinadas estructuras de datos como los arreglos uni o bi dimensionales.

- Falta de apropiación por parte del alumno de la metodología de resolución de problemas propuesta.
- Escasa experiencia en el manejo del lenguaje de programación utilizado en la asignatura.
- Escasa integración de la asignatura con las otras materias de la carrera, especialmente con Lógica Matemática Elemental y con Introducción al Álgebra.

Fundamentación de la innovación

La asignatura Introducción a la Algorítmica y Programación tiene como objetivo iniciar al alumno en el desarrollo de algoritmos resolutivos de problemas que pueden ser procesados por computadora. Esta materia es una de las bases fundamentales de la currícula de las carreras de Analista en Computación, Licenciatura y Profesorado en Ciencias de la Computación. A través de los años desde la creación de estas carreras se han introducido diversas modificaciones en la enseñanza de la algorítmica. Aún hoy en día se puede verificar que no hay un consenso en los métodos a utilizar. Esto fue corroborado mediante un relevamiento de los distintos programas de enseñanza de la algorítmica en distintas universidades de la Argentina y de otros países (Internet mediante), así como también del contacto con profesores responsables de materias afines en otras universidades del país y del exterior. Una de las razones para que esto sea así es que no existe un único método para la resolución de algoritmos y en la actualidad este tema es un campo de estudio e innovación permanente.

En el caso específico de nuestra asignatura y con relación a su dictado en años anteriores entendemos que existen aspectos que requieren de cambios y que se enmarcan en las innovaciones a practicar en la enseñanza de la algorítmica, esos aspectos son:

- 1- El método de desarrollo de algoritmos que se viene aplicando:

Identificamos en éste método, en particular algunos de los pasos que lo componen, un grado de dificultad que supera el del propio problema que se pretende resolver. El método consiste en especificar estados, que son predicados que se evalúan aplicando las reglas de lógica. La escritura de estos estados permite en un segundo paso la deducción de las acciones que dan forma al algoritmo. La especificación debe ser lo más precisa posible para que se le puedan aplicar las reglas de lógica y conseguir así su simplificación. En determinados problemas resulta para los alumnos más fácil deducir la acción directamente que especificar el estado, esto es más evidente en el caso de datos estructurados como son los arreglos, que tienen la particularidad de ser complejos en su escritura por que necesitan de variables índices para ser explicados. Otra situación en la que el método presenta inconvenientes es en la deducción de las partes componentes de un ciclo. El método exige la escritura de un predicado invariante y a partir de él se construyen los estados que permitirán derivar las acciones buscadas. Hemos detectado que en muchos problemas las acciones se pueden deducir de manera prácticamente trivial sin conocer el invariante y en otros problemas el invariante resulta muy difícil de encontrar o muy difícil de especificar (especialmente en el caso de los arreglos que antes comentamos). No existen reglas que permitan deducir de una manera sistemática a los invariantes. En el caso que el predicado invariante hubiera sido hallado de pueden especificar un conjunto de estados a partir de los cuales se deducen las acciones buscadas.

En este punto la innovación a implementar consiste en simplificar los pasos del método de desarrollo, modificar la manera de hallar las acciones que corresponden a un ciclo, no hacer depender exclusivamente del invariante la deducción de las acciones inherentes al ciclo sino dejar abierto la posibilidad de escribir las acciones aún cuando el invariante no ha sido formalmente especificado. Esta innovación que consiste en una modificación del método anterior tiene como fin contrarrestar los problemas detectados, descritos anteriormente y permitirá por su menor cantidad de pasos tratar una mayor cantidad de problemas.

2- La cantidad de ejercicios que se realizan en las clases prácticas:

El procedimiento descrito en párrafos anteriores, si bien por una parte tiene consistencia lógica, por otra exige una dedicación de tiempo muy alta, al punto que en una clase ordinaria en el mejor de los casos se pueden realizar a lo sumo tres ejercicios. Esto motivó que las prácticas se redujeran en cuanto al número de ejercicios y además el tipo de ejercicios que se incluyeron, fueran de los más simples, precisamente para economizar tiempo de resolución. Esto tuvo como resultado final una actividad práctica más acotada, un peso más alto en el tiempo dedicado a los aspectos teóricos y menos tiempo dedicado a los aspectos prácticos, en definitiva una menor práctica concreta. Esto también se verificó en las actividades de laboratorio, práctica en computadoras, la cuál históricamente empezó en nuestra asignatura a fines del primer cuatrimestre. Respecto a la traducción al lenguaje Pascal de los algoritmos desarrollados en las clases prácticas una cuestión que fue descuidada fue el estilo de programación y que hace a la legibilidad de los programas, aspecto que se relaciona con la corrección de programas, con su reutilización, con la posibilidad de compartir entre programadores sus producciones.

Las innovaciones que se proponen en este sentido son: mayor cantidad de ejercicios por práctica, ejercicios de complejidad creciente, utilización de un conjunto de convenciones de escritura previamente establecidas y un contacto con computadoras más temprano en el proceso de aprendizaje. En cuanto a la escritura de los algoritmos en un lenguaje de programación concreta (generalmente Pascal) se exigirá el cumplimiento de convenciones de escritura previamente establecidas y se destacará en su aplicación las ventajas de estas prácticas. Con estas medidas se pretende alcanzar una práctica más intensa, mayor experiencia en el desarrollo de algoritmos por parte de los alumnos de forma que arriben al proyecto final de cierre de la materia con mayores habilidades y conocimientos prácticos para realizarlo. Adicionalmente la utilización de las convenciones de escritura facilitará el análisis de los códigos escritos por los alumnos y permitirá profundizar la corrección de los mismos para detectar con mayor facilidad cuestiones teóricas implícitas en aquellos.

3- Escasa integración con otras materias y desfase en la aplicación de conocimientos de otras materias en algorítmica:

La integración a que se hace referencia es en sentido horizontal es decir entre las materias del primer año. La integración vertical, con las materias de los años subsiguientes en la currícula está cubierta con los contenidos de la materia, no obstante es una cuestión que debe ser ajustada y que está incluido en los asuntos a tratar en otros trabajos. Respecto a las materias del primer curso es posible la integración de conocimientos mediante la incorporación de una ejercitación adecuada. Con respecto a los conocimientos sobre los formalismos lógicos, estos son adquiridos por los alumnos en la materia Lógica Matemática Elemental, correspondiente al primer cuatrimestre de primer año, es decir, en paralelo al cursado de la asignatura Introducción a la Algorítmica y Programación (IAyP)

que se extiende durante el primer y segundo cuatrimestre de primer año. La experiencia de los últimos años ha demostrado que debe existir una necesaria maduración de los conocimientos de lógica antes de ser utilizados en IAyP, más aún cuando los mismos no son obligatorios para el cursado de la asignatura, es decir no hay correlatividades entre ambas.

Las innovaciones en este sentido son coordinar la aplicación de los conocimientos de lógica en la medida que son asimilados por los alumnos, para ello se prevé acordar la incorporación de conocimientos en la medida que son adquiridos. Se prevé la realización de reuniones de coordinación con los docentes de las asignaturas involucradas con el objeto de consensuar el cronograma de dictado de contenidos.

- 4- La superposición de actividades entre materias distintas en el primer año de las carreras de computación:

Por último, creemos que un aspecto didáctico muy importante en la construcción de programas es la realización de proyectos. Estos consisten en resolver un problema real de cierta complejidad. Estos proyectos habían perdido importancia en su utilidad didáctica puesto que en el final del primer cuatrimestre los alumnos aún no habían madurado los conceptos, y al final del segundo cuatrimestre el proyecto final se solapaba con los últimos parciales y recuperatorios de las demás materias de primer año, es decir con Cálculo I e Introducción al Álgebra. A esto se sumaba que muchos alumnos no podían cumplir con los plazos del proyecto y no tenían suficiente tiempo de estudio para el resto de las materias. Como consecuencia de ello un cierto porcentaje de alumnos quedaban libres en esta última etapa.

En este sentido la innovación que se propone es adelantar dos semanas el comienzo del cursado de la materia Introducción a la Algorítmica y Programación en el segundo cuatrimestre con el objetivo de finalizar el cursado de la misma dos semanas antes y facilitar el armado de un cronograma más holgado para la entrega del proyecto final y evitar la superposición con la preparación de los exámenes y recuperatorios.

Objetivos de la innovación

El proyecto que pretendemos llevar adelante persigue como objetivos superar aquellos problemas detectados y que se han comentado en el punto anterior, sistematizar los cambios propuestos y permitir realizar su posterior evaluación, a fin de permitir el ajuste o corrección de los cambios propuestos.

Las innovaciones a practicar tienen como objetivos específicos:

- Mejorar la destreza de los alumnos para solucionar problemas de cierta complejidad mediante algoritmos correctos, que puedan ser traducidos a un lenguaje de computadora y ser ejecutados en ella.
- Aumentar la cantidad de ejercicios realizados por alumno en cada unidad temática y al término de cada cuatrimestre.
- Aumentar el porcentaje de alumnos que habiendo agotado todas las instancias evaluativas regularizan la asignatura, medido con respecto al año anterior.

- Aumentar el porcentaje de ejercicios que se realizan en cada clase práctica con respecto a la cantidad que se realizaban a igual unidad temática el año anterior y aumentar la media de ejercicios que se hacen por clase a lo largo del cuatrimestre.
- Mejorar el grado de apropiación del método de desarrollo de algoritmos por parte de los alumnos.
- Aumentar el tiempo de contacto de los alumnos con la computadora y anticipar dicho contacto con relación al momento en que se hacía anteriormente.
- Incorporar buenos hábitos de estilo en la escritura de programas.
- Mejorar la interrelación de la materia con las otras materias de la currícula.

Descripción de la innovación

La innovación que se intenta llevar a delante involucra un conjunto de estrategias y prácticas que se pueden resumir en:

- Rediseñar el material teórico y orientarlo hacia un enfoque no ligado a la derivación de programas a partir de la especificación sino hacia el uso del método descendente y la modularización como ejes para la resolución de problemas.
- Dotar a las guías de trabajos prácticos de mayor cantidad de ejercicios y de complejidad creciente.
- Aplicar en las primeras tres unidades temáticas un método informal de resolución de problemas, el objetivo es que el alumno adquiera la destreza en la resolución de problemas simples, que se pueden realizar en unos pocos pasos, sin descuidar aspectos teóricos como el análisis previo de cuales son los datos de entrada necesarios, cuales con los resultados que se buscan y cuales son los conocimientos previos que se necesitan para abordar el problema.
- Introducir en la cuarta y quinta unidad temática un sistema de especificación semiformal de problemas que permita la construcción de predicados a fin de completar la etapa de análisis de un problema pero con mayor precisión.
- Se Introducirá en la sexta unidad temática un método para la solución de problemas que tratan secuencias y se permitirá la escritura directa de las acciones sin (escribir previamente su pre y poscondición) cuando estas resulten evidentes o triviales.
- A continuación se desarrollarán en la asignatura el resto de temas que comprende aplicando esta metodología.
- Paralelamente al desarrollo de los aspectos teóricos-prácticos que se desarrollan en aula, se introducirá la traducción a un lenguaje de programación de los algoritmos y su ejecución en computadora, se prevé que esto se haga a partir de la segunda semana de cursado de la asignatura en una clase opcional para el alumno.
- Por cada unidad temática se solicitará a los alumnos, de manera individual y/o grupal, un ejercicio resuelto codificado en Pascal. Este programa será corregido por el docente de prácticos conjuntamente con sus ayudantes y devuelto al alumno con las observaciones que resulten de dicha corrección. Se evaluará especialmente el cumplimiento de las convenciones adoptadas para la escritura de programas en Pascal.
- Los docentes contarán con un plan de ejercicios mínimos a resolver, estratégicamente elegidos por su completitud y complejidad. Estos ejercicios serán realizados en clase y resueltos con los alumnos.
- Las comisiones no superarán los 30 alumnos, para garantizar una atención individual acorde a las necesidades del curso.
- Se destinará una comisión exclusivamente para alumnos recursantes a fin de trabajar de manera específica con ellos los temas cuyo enfoque haya variado con respecto al año anterior.

- Se articulará IAyP con las asignaturas del mismo ciclo de enseñanza de manera que se pueda alcanzar una equilibrada integración de los temas, en particular la construcción de predicados lógicos, la creación y utilización de funciones.
- Se realizará una reunión de cátedra semanal para intercambiar impresiones sobre la marcha del dictado de clases, para evaluar la respuesta de los alumnos a las ejercitaciones propuestas y para ir ajustando la metodología que se pone en práctica en cada unidad temática.

La modalidad de trabajo comprende:

- Cada semana los alumnos asistirán a una clase teórica de tres horas dónde se desarrollan los conceptos necesarios para abordar la resolución de los trabajos prácticos. Además dispondrán de una clase de repaso (no obligatoria) de igual duración y una clase de consulta. En la clase de repaso podrán solicitar el desarrollo pormenorizado de los temas que no fueron entendidos en la clase teórica.
- Por semana los alumnos tendrán dos clases prácticas de dos horas de duración y una clase de repaso-implementación (no obligatoria) de igual duración. En la clase de repaso-implementación se podrán retomar ejercicios que quedaron pendientes o no se entendieron así como también implementar en computadora los algoritmos solicitados por el docente, esta actividad se realizará en Laboratorio y los alumnos contarán con la asistencia del docente encargado de prácticos y de al menos un ayudante alumno. En las clases prácticas el docente planteará el objetivo de la práctica, desarrollará si son necesarios algunos conceptos mínimos, para complementar lo explicado en las clases teóricas, y pautará el orden y los ejercicios que como mínimo deberán resolverse durante la clase. Con la ayuda de ayudantes alumnos (seleccionados mediante concurso) el docente irá evacuando las consultas que vayan surgiendo, haciendo aclaraciones a toda la clase cuando fuera pertinente. Desarrollará ejercicios completos a modo de ejemplo y discutirá diversas soluciones para un mismo problema con los aportes que realicen los alumnos de sus propios resultados.
- En las clases practicas en laboratorio se utilizará la computadora, los alumnos traducirán al lenguaje Pascal los algoritmos desarrollados oportunamente. El docente con los ayudantes alumnos irá evacuando las consultas que vayan surgiendo, haciendo aclaraciones a toda la clase cuando fuera pertinente.
- Una vez por semana se realizarán reuniones entre el docente responsable de la asignatura, a cargo de la teoría, y los docentes responsables de las clases prácticas para coordinar las actividades, acordar ajustes en la metodología, evaluar la marcha en el dictado de las clases prácticas y realizar los ajustes que se acuerden.
- Una vez por semana se realizarán reuniones entre el docente responsable de cada comisión de práctico y los ayudantes alumnos cargo para acordar la forma de realización de los ejercicios prácticos.
- La evaluación del aprendizaje consistirá en dos exámenes parciales y cada uno podrá recuperarse una única vez. Al término del primer cuatrimestre se exigirá el desarrollo de una solución en todas sus fases: algoritmo, implementación en computadora, manual de usuario e informe técnico. Al final del curso se propondrá un problema que podrá ser resuelto por grupos de hasta tres alumnos.

Investigación Evaluativa

Complementariamente al Proyecto pedagógico presentado se llevará adelante una investigación evaluativa. Las innovaciones que se pretenden implementar requieren de una evaluación objetiva de los métodos propuestos. Es preciso establecer la innovación y los cambios que

provocan en el proceso de enseñanza. Sin esta evaluación no se tiene la necesaria sistematización en la aplicación de cambios que permita reconocer el impacto producido, es decir el antes y el después. Por lo tanto la investigación debe ser incorporada como una etapa complementaria necesaria que permita decidir en última instancia cuales son los resultados, cuales pueden ser las fallas y dónde deben realizarse las correcciones.

Los problemas que se quieren abordar implican no solo la evaluación de resultados, entendiendo estos como alcance de conocimientos adquiridos por los alumnos, sino además la evaluación de los procesos a que da lugar la implementación de la innovación propuesta. Esto último implica revisar la dinámica de las clases, el comportamiento del alumnado, su respuesta e interés, la dedicación puesta de manifiesto, la actitud de los docentes frente a las situaciones que se plantean como problemáticas, la identificación de situaciones comunes entre las distintas comisiones, el grado de dificultad que demuestran los alumnos frente a determinados temas, la discusión intra-cátedra para establecer un consenso en la mejor manera de introducir o dictar un tema.

La investigación implica una revisión continua de los procesos y el ajuste según la percepción del equipo de docentes, la detección de las deficiencias en el aprendizaje suele en determinados caso una percepción que se obtiene de la marcha de los trabajos prácticos, del tipo de preguntas que los alumnos realizan en clase, pero muy especialmente de los exámenes parciales dónde quedan plasmados uno de los aspectos más relevantes como son los resultados concretos que cada alumno obtiene.

El conjunto de actividades ligadas a esta etapa están en plena elaboración y constituyen unos de los productos de la innovación.

Conclusiones

El proyecto de Innovación pedagógica que presentamos tiene como objetivo la mejora del proceso de enseñanza aprendizaje, recoge la experiencia acumulada y reconoce de paso falencias detectadas en años anteriores. Constituye un esfuerzo por parte de la cátedra en sistematizar y profundizar el mejor conocimiento de los procesos involucrados en el aprendizaje de la algorítmica y en su enseñanza. No se agota solo en enseñar sino que además se compromete en la evaluación de ese proceso, de manera que se obtiene aquí un producto de esta experiencia que se podrá compartir y que se aspira permita enriquecer el trabajo que año a año se desarrolla. Resulta evidente que algunas de las situaciones problemáticas planteadas escapan a las cuestiones pedagógicas desarrolladas y reconocen causas que pueden ser absolutamente exógenas a ellas, (por ejemplo el porcentaje de alumnos que no agotan todas las instancias evaluativas), no obstante entendemos que está en la base de los posibles caminos para mejorar la retención de alumnos sin menoscabar la calidad de la enseñanza, explorar nuevas alternativas y si es necesario volver a prácticas que se habían dejado de lado. A la vez que se intentan mejoras es necesario sistematizar estos esfuerzos de manera que podamos medirlos y compararlos para su corrección en caso que fuera necesario, aparece así la investigación evaluativa como una instancia necesaria y vital que complementa la innovación.

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] A ROSSO, M. DANIELE. Algunos errores sistemáticos detectados en el proceso de aprendizaje de la Algoritmica´. 4to. Ateneo de Profesores Universitarios de Computación y Sistemas. San Luis Noviembre de 1996.
- [4]- P.C. SCHOLL - J.P. PEYRIN. Esquemas Algorítmicos Fundamentales. Secuencias e Iteración. Ed. Masson, 1991.
- [5]- NIKLAUS WIRTH. Algoritmos + Estructuras de datos = Programas. Ed. del Castillo S.A., 1980.