Programación y Algoritmos: Análisis y Evaluación de Cursos Introductorios

Sandra Casas y Verónica Vanoli

Universidad Nacional de la Patagonia Austral. Unidad Académica Río Gallegos
Lisandro de la Torre 1070. CP 9400. Río Gallegos, Santa Cruz, Argentina
Tel/Fax: +54-2966-442313/17.
E-mail: {lis, vvanoli}@uarg.unpa.edu.ar

Resumen

La identificación de los obstáculos en el proceso de enseñanza-aprendizaje de la programación y sus algoritmos permite orientar las nuevas propuestas académicas y didácticas, para que la planificación y gestión de los recursos y materiales sea de mayor utilidad para el alumno. El estudio y análisis de las problemáticas del proceso se constituye en el punto de partida. En este trabajo se presentan los resultados de los primeros estudios realizados en la carrera Analista de Sistemas, de la Unidad Académica Río Gallegos (UARG), de la Universidad Nacional de la Patagonia Austral (UNPA).

1. Introducción

A lo largo de la historia de educación en ciencias de la computación o informática, la estructura de los cursos introductorios ha sido un tema de intenso debate. Una de las cuestiones mas discutidas es el papel de la programación en la fase introductoria del plan de estudios. Lo cierto es que prácticamente desde los comienzos de la disciplina [1] [2] los cursos introductorios de informática se han enfocado principalmente en el desarrollo de las capacidades de la programación. A pesar de las falencias observadas, los beneficios y necesidades parecen ser mayores [3]. Esto principalmente se debe a que se consideran pre-requisito fundamental para cursos de nivel intermedio y nivel avanzado. De esta forma el alumno inicial en primera instancia debe adquirir este tipo de conocimiento y capacidades.

La Programación es el arte y la técnica de construir y formular algoritmos de una forma sistemática [4]. Este concepto involucra un "don" y un "procedimiento", tal vez por estas características aprender a desarrollar algoritmos y programas aunque se considere una capacidad básica no es una tarea simple, trivial y fácil para el alumno. Un extenso trabajo dirigido a mejorar el proceso de enseñanza-aprendizaje referente a la programación de algoritmos da cuentas de lo complejo que puede resultar abordar este tipo conocimientos y habilidades, principalmente en el alumno novato que se inicia sin ningún tipo de conocimientos previos relacionados. Por estos motivos, docentes e investigadores buscan nuevas estrategias didácticas y pedagógicas, herramientas y recursos que contribuyan a que este proceso sea más exitoso [5] [6] [7] [8] [9] [10].

Asimismo, los fenómenos de deserción y desgranamiento que se producen en los primeros años de las carreras de informática provocan necesariamente una mirada analítica, reflexiva y superadora. La búsqueda de propuestas de cambios y mejoras es un objetivo, pero debe surgir de una clara identificación de los principales problemas y dificultades que se presentan en el proceso de enseñanza-aprendizaje. Por eso, en primera instancia se debe identificar, analizar y estudiar los factores que plantean obstáculos en el proceso. En particular, se trata de una serie de elementos a analizar: la complejidad de los contenidos conceptuales, la dificultad de aplicar dichos contenidos en la práctica, la efectividad y utilidad de los recursos y materiales empleados, además de las estrategias didácticas que se ponen en juego. A partir de esta información, puede reestructurarse el proceso completo o simplemente hacer cambios parciales. En este sentido, en este trabajo se presentan los resultados y análisis de los primeros estudios realizados en la carrera Analista de Sistemas, de la UARG - UNPA.

2. Estudio: Aspectos de la evaluación

El estudio se realizó a través de un cuestionario utilizado previamente por [11], aunque se introdujeron algunas variantes para hacerlo mas especifico al contexto particular de estudio. Lo interesente del mismo es que permite evaluar aspectos independientes al paradigma y al lenguaje de programación aplicado.

El cuestionario se compone de cuatro secciones. La primera sección recolecta información general del alumno. La segunda sección esta dirigida a identificar los obstáculos en el aprendizaje, relacionados a los contenidos conceptuales y las capacidades y habilidades que deben ser adquiridos. A continuación se presentan las consignas e ítems correspondientes al estudio de la segunda sección:

¿Qué tarea le resultó difícil mientras aprendía programación?
(A) Usar entorno de desarrollo.
(B) Aprender la sintaxis del lenguaje.
(C) Diseñar un programa para resolver una tarea.
(D) Entender estructuras de programación.
(E) Dividir la funcionalidad en subprogramas.
(F) Realizar la traza o prueba de escritorio manualmente.
(G) Encontrar sus propios errores.
(H) Corregir sus propios errores.

¿Qué concepto de programación ha sido más difícil de aprender?
(A) Selección.
(B) Iteración.
(C) Arreglos – Vectores.
(D) Arreglos – Matrices.
(E) Arreglos – Ordenamientos.
(F) Arreglos – Búsquedas.
(G) Modularidad – Funciones / Funciones y Procedimientos.
(H) Modularidad – Unidades / Clases.
(I) Parámetros.
(J) Recursividad.
(K) Punteros – Referencias.
(L) Pilas – Colas.
(M) Listas.
(N) Ficheros.
(O) Usar librerías / APIs.

Cada ítem fue evaluado por la dificultad de aprendizaje mediante cinco opciones: muy fácil, fácil, medio, difícil y muy difícil.

La tercera sección tiene el propósito de evaluar aspectos relacionados a los materiales y a las situaciones de aprendizaje. Aquí la evaluación del alumno implica elegir entre las opciones: nada, poco, medio y mucho. Las consignas e ítems en que se estructura la tercera sección son las siguientes:

¿Cuándo cree que aprende programación?
(A) Cuando lee libros de programación.
(B) Cuando lee el apunte.
(C) En la clase de teoría.
(D) En la clase de práctica.
(E) Cuando estudia/practica solo.
(F) Cuando estudia/practica en grupo.
(G) Cuando practica en papel.
(H) Cuando practica en la computadora.

¿Qué material es de ayuda para aprender programación?
(A) Libros de programación.
(B) Apunte de cátedra.
(C) Ejercicios de ejemplo.
(E) Trabajos prácticos.
(F) Internet.

La última sección esta destinada a que el alumno exprese sugerencias y/o críticas que aporten información al estudio no contemplada.

La experiencia se realizó con alumnos que regularizaron los dos cursos iniciales de programación (asignatura “Programación I” del primer año y asignatura “Programación II” del segundo año). En el estudio participaron alumnos de distintos años de cursado, característica que permite detectar problemáticas más generales y estructurales. Para el cuestionario participaron un total de 51 alumnos, de los cuales el 43% eran de sexo femenino y el 57% de sexo masculino.

3. Análisis de Resultados

En la Figura 1 se observa que las tareas para aprender programación tienen para el alumno en general una dificultad de fácil a media. Las tres tareas más difíciles: (F) “Realizar la traza o prueba de escritorio manualmente”, (G)

"Encontrar sus propios errores" y (H) "Corregir sus propios errores", están estrechamente relacionadas, siendo G y H consecuencia de F.

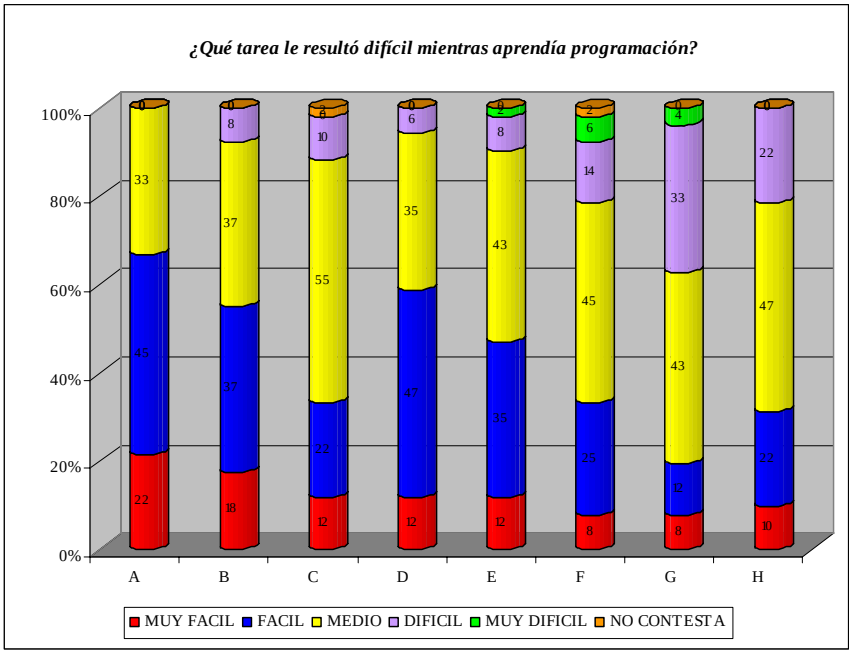

Figura 1: Dificultad de la tareas.

En la Figura 2 se observan que el contenido conceptual más difícil es Recursividad (J), dónde el 43% de los alumnos marcó como un concepto difícil y el 24% como muy difícil. Punteros – referencias (K), listas (M) y Ficheros (N), serían los otros conceptos menos accesibles. Es notable que más del 80% de los alumnos ha considerado que los conceptos Selección (A), Iteración (B) y Vectores (C) son de muy fácil a fácil.

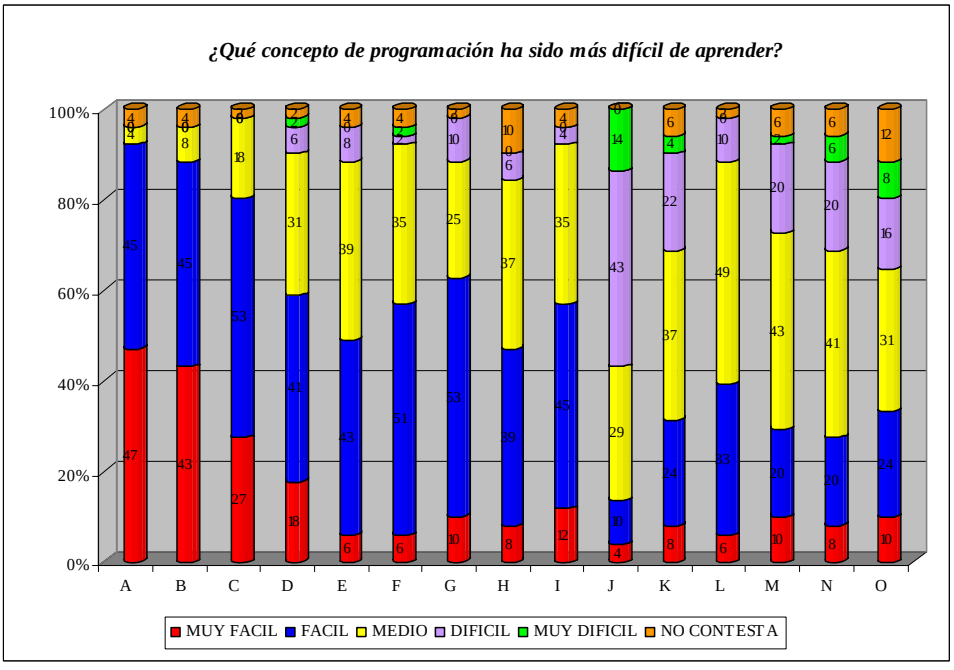

Figura 2: Dificultad de los contenidos conceptuales.

En la Figura 3 se observa que los alumnos consideran que aunque la lectura de libros y apuntes de cátedra resultan útiles para aprender, resultan mas eficaces los encuentros en las clases ya sea de teoría como de práctica. Con esto queda de manifiesto que el alumno es renuente a la lectura, aún más si el texto del material es en inglés. Por otro lado, el estudio individual le resulta más productivo que en grupo. Las prácticas en computadora tienen mucho más valor que las prácticas en papel (la diferencia es notable en las respuestas).

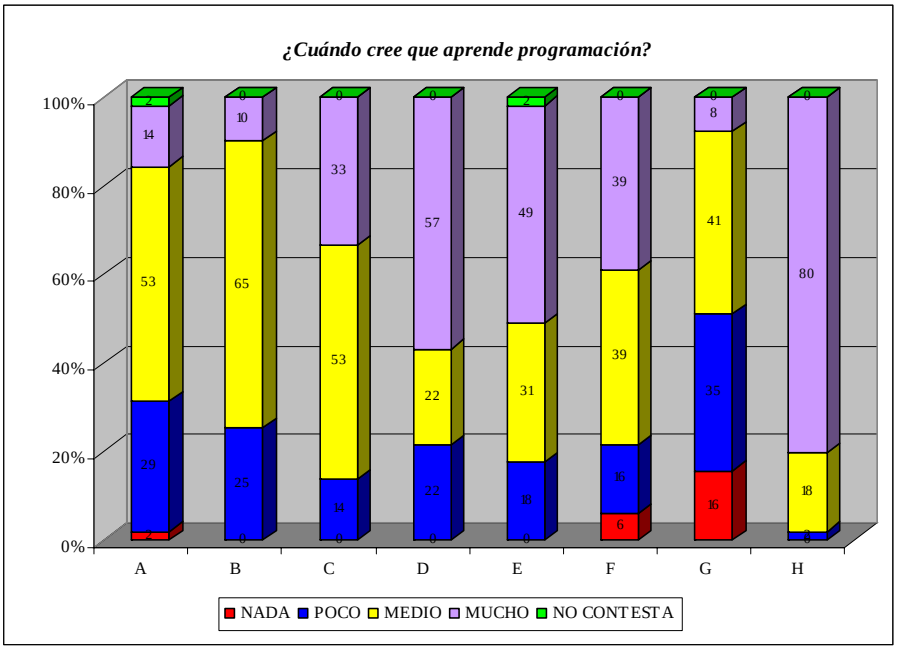

Figura 3: Evaluación de las situaciones de aprendizaje.

En la Figura 4 se observa que los ejercicios de ejemplo y los trabajos prácticos son de mayor ayuda para los alumnos que el material teórico (libros y apuntes), lo que se corresponde con los resultados obtenidos en la consigna anterior.

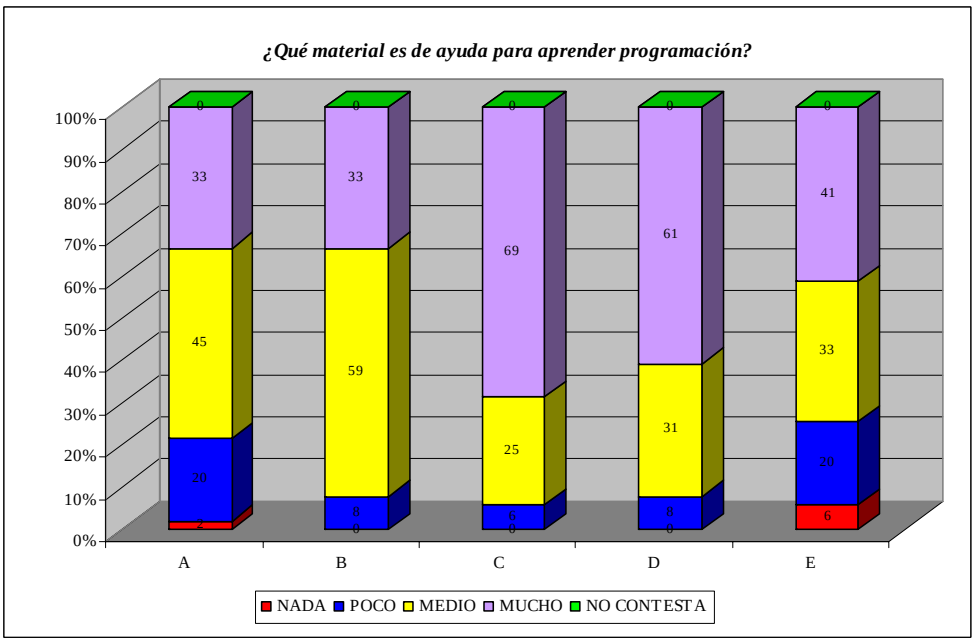

Figura 4: Evaluación de recursos y materiales.

4. Estudios Relacionados

El Plan de Actividades de Apoyo Académico al Ciclo Básico, dependiente del Programa de Acceso y Permanencia de la Unidad Académica, en el año 2005 realizó diversos estudios a los alumnos ingresantes de todas las carreras. Del procesamiento y análisis de los datos obtenidos de las actividades de diagnóstico inicial y diagnóstico psicopedagógico, el informe concluye: "Del análisis de las pruebas psicopedagógicas suministradas a los alumnos ingresantes a la carrera de Analista de Sistemas, podemos señalar que mas de la mitad de los mismos demostraron un nivel intelectual que responde a la media o superior a ella (Resultados del Test Domino D-48: 48% Término medio, 9% término medio-alto, 34% Término medio-bajo, 9% término bajo). Respecto a la capacidad combinatoria y por ende, del razonamiento hipotético deductivo es alcanzado por un 25% de los sujetos, pero un número importante se encuentra aún en el periodo de transición (Resultados del Test Diagnóstico-Operatorio: 29% Pensamiento Formal, 14% Pensamiento Concreto y 57% Pensamiento en Transición) en vías de adquirir las mismas. En este sentido, y teniendo en cuenta la carrera elegida, se estima que las capacidades propias del pensamiento formal son necesarias al momento de cursar asignaturas del área de las Matemáticas, por lo cual este grupo debería encontrar, o ser ayudado a encontrar, las estrategias necesarias que estimulen las capacidades para acceder a la etapa de pensamiento mencionada. Se plantea *ser ayudado*, ya que un número considerable de alumnos no reconoce poseer algún déficit en su formación que pueda ser obstáculo en el proceso de enseñanza-aprendizaje emprendido".

5. Conclusiones y Acciones futuras

Los resultados obtenidos han servido para re-estructurar las asignaturas en cuanto a la organización y distribución de los contenidos, como a su disposición en el plan de estudios. En cuanto a las observaciones más puntuales, una serie de estrategias y actividades se proponen para aplicar en forma interna: (i) Intensificar las prácticas de traza y de esta forma mejorar la búsqueda y corrección de errores propios. (ii) Disponer más tiempo en la planificación, para trabajar los contenidos más difíciles: Recursividad, Punteros, Listas y Ficheros. (iii) Proponer actividades prácticas grupales en computadora, al estilo programación por parejas.

6. Referencias

[1] ACM Curriculum Committee on Computer Science. Curriculum '68: Recommendations for the undergraduate program in computer science. Communications of the ACM, 11(3):151-197, March 1968.

[2] ACM Curriculum Committee on Computer Science. Curriculum '78: Recommendations for the undergraduate program in computer science. Communications of the ACM, 22(3):147-166, March 1979.

[3] Computing Curricula 2001. Computer Science. Final Report (December 15, 2001). IEEE & ACM.

[4] Wirth N. "Algoritmos e Estruturas de Dados". LTC Informática-Programação, 1989.

[5] Azeredo P. A. "Uma proposta de Plano Pedagógico para a Matéria de Programação". Anais do II Curso: Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI). Editora Universitária Champagnat. 2000.

[6] Soares T. C. A. P., Cordeiro E. S., Stefani Í. G. A., Tirelo F. "Uma Proposta Metodológica para o Aprendizado de Algoritmos em Grafos Via Animação Não-Intrusiva de Algoritmos". III Workshop de Educação em Computação e Informática do Estado de Minas Gerais 2004. Brasil.

[7] Cortes E., Vanoli V., Casas S. "BigBang: un recurso didáctico-pedagógico en el aprendizaje de la implementación de algoritmos en pseudocódigo". VIII WICC. Universidad de Morón. Buenos Aires. Junio 2006. ISBN 978-950-9474-35-2.

[8] Azul A. A., Mendes A. J. EDDL: "Um Programa Didático sobre Estruturas de Dados Dinâmicas Lineares". 3º Simpósio Investigação e Desenvolvimento de Software Educativo – 1998. Évora, Portugal.

[9] Black, P. E. "Dictionary of Algorithms and Data Structures". NIST 2005.

[10] Garcia, I. C., Rezende, P. J., Calheiros, Astral F. C. "Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos". Revista Brasileira de Informática na Educação (RBIE) nº 01. 1997.

[11] Lahtinen E., Ala-Mutka K., Jarvinen H. "A Study of the Difficulties of Novice Programmers". ITiCSE 2005, Portugal.