

ESTRATEGIAS METACOGNITIVAS PARA EXPLICAR LA COMPLEJIDAD DEL DISEÑO ALGORÍTMICO EN ESTUDIANTES INICIALES DE PROGRAMACIÓN

Marcia Mac Gaul; Marcela F. López; Paola del Olmo; Eduardo F. Fernández

Consejo de Investigación de la Universidad Nacional de Salta (CIUNSa.)

1. INTRODUCCIÓN

Este trabajo relata una indagación realizada a estudiantes iniciales de Programación, de la Facultad de Ciencias Exactas de la Universidad Nacional de Salta, en el marco del Proyecto de Investigación CIUNSa. N° 2154, denominado *Estrategias Didácticas apoyadas por Tecnología, tendientes a reducir índices de deserción en el primer año de carreras informáticas*, acreditado en el Consejo de Investigación de la mencionada universidad, para su desarrollo entre 2014 y 2017.

La investigación indaga las trayectorias educativas de los estudiantes, con el objeto de descubrir dificultades que conllevan recurrentemente al abandono, sobre todo aquellas vinculadas con errores cognoscitivos sistemáticos en la lógica algorítmica.

El aprendizaje de la Programación es un proceso complejo para los estudiantes universitarios iniciales, ya que requiere de la apropiación de estrategias específicas de abordaje y resolución que en muchos casos, no han sido afrontadas desde los contenidos previos. La resolución de problemas computacionales involucra un proceso de abstracción reflexiva, sostenido por procesos lógicos matemáticos que se adquieren en interacción con objetos específicos, contruidos en una dialéctica teoría – práctica, que orienta las decisiones estratégicas a seguir en cada diseño algorítmico. La imposibilidad de superar estas dificultades, genera en los estudiantes una sensación de fracaso y frustración y un consecuente abandono de la carrera al no poder alcanzar los objetivos deseados por él y esperados por sus docentes.

Los estudiantes de primer año, de las carreras Licenciatura en Análisis de Sistemas (LAS) y Tecnicatura Universitaria en Programación (TUP), cursan en el primer cuatrimestre la asignatura Elementos de Programación. El alumno que concurre es, en su gran mayoría, un egresado reciente del nivel anterior. No siempre dispone de procesos cognitivos que le permita un grado de abstracción adecuado para permanecer en el primer año. Tampoco cuenta con hábitos ni estrategias de estudio que le permita consolidar su inclusión en el sistema superior.

El grupo de investigación coincide que hablar de fracaso en los aprendizajes en el Nivel Superior, conduce a orientar la mirada hacia la Evaluación. Cuestionarse acerca de cómo trabajar sobre las potencialidades y posibilidades y no sólo sobre lo que los sujetos no saben, o lo que desde las Cátedras se marcan como deficiencias.

Esto significa, no solo privilegiar la transmisión de los conceptos disciplinares sino también explicitar las estrategias pertinentes para su aprendizaje. Es decir, trabajar lo metacognitivo, puesto que verdaderamente pretendemos mejorar en los estudiantes sus procesos de elaboración conceptual, así como el aprendizaje de las estrategias cognitivas que favorezcan el desarrollo progresivo de su autonomía en la construcción de conocimientos. El auténtico aprendizaje (con comprensión) supone espacios para pensar y hacer, espacios que brinden posibilidades para discutir, argumentar y dar fundamentos, de modo que el aprendizaje se oriente a la construcción de sistemas de

pensamiento. Esto exige un cambio radical: pasar de la adquisición de información a la comprensión.

La cátedra de Elementos de Programación está integrada por dos profesores que imparten clases teóricas y nueve auxiliares docentes dispuestos en comisiones de clases prácticas.

Los contenidos de Programación se evalúan en dos parciales y sus respectivas recuperaciones. Estas instancias remediales se aplican al final del cuatrimestre, buscando lograr que los alumnos tengan oportunidad de alcanzar un mayor nivel de comprensión del contenido a través de prácticas extras en el diseño algorítmico. Para este fin, la cátedra ofrece un Taller de Diagramación Algorítmica, de cursado semanal voluntario, apoyado por tutoría presencial y a distancia.

El artículo presenta resultados correspondientes a una evaluación parcial de Diagramación Algorítmica, aplicado a los dos meses de iniciado el cursado.

2. OBJETIVOS

La indagación se centra en dos aspectos, 1) el grado de dificultad que el alumno asigna a cada problema, como punto de partida para estudiar su posible relación entre esta valoración y la solución algorítmica que realiza; 2) Desde la palabra del docente, el grado global de complejidad que cree que cada uno de estos problemas presenta para sus estudiantes y la ponderación de cada aspecto que contribuye a su complejidad.

3. METODOLOGÍA

Se realiza una indagación cualitativa de tipo exploratoria descriptiva, que permita un acercamiento a los procesos y estrategias que sostienen los estudiantes.

3.1. Instrumento

Como instrumento metodológico se utiliza el segundo parcial de Elementos de Programación. La evaluación es escrita. Se aplica para un desarrollo de tres horas como máximo. Abarca contenidos de Algoritmos cuya solución demanda el uso de variables simples, variables indizadas unidimensionales y bidimensionales. Integra algoritmos fundamentales de ordenamiento y búsqueda. Consta de tres problemas cuya solución debe presentarse usando Diagramas N-S¹, como herramienta de especificación algorítmica.

3.2. Población estudiantil

La población de 118 estudiantes corresponde a la asistencia registrada en el segundo parcial de la asignatura.

Se solicita a los alumnos que valoren la formulación de cada problema según la escala: Muy fácil – Fácil – Moderado – Difícil – Muy difícil.

3.3. Población docente

Los 11 sujetos de la muestra integran el equipo docente de la cátedra Elementos de Programación. Dos de ellos, con cargos de Profesor, imparten las clases teóricas; siete con cargos de Jefe de Trabajos Prácticos, imparten clases prácticas y dos con cargos de

¹ Nassi Shneiderman Diagrams.

Auxiliares Docentes de Segunda Categoría (alumnos avanzados), colaboran con los Jefes en las clases prácticas.

Los docentes poseen vasta experiencia en las asignaturas iniciales de Programación, con un promedio de 16 años de antigüedad. El 72% de los miembros de la cátedra integra el equipo de investigadores del Proyecto actual y anteriores, desde el año 2007. Se trata de un grupo cohesionado alrededor de procesos investigativos y de desarrollo de materiales educativos, para la enseñanza de la Programación, en los cursos básicos universitarios. En este contexto, la pedagoga del Proyecto desarrolla talleres de formación docente centrados en diversos temas, entre los que se encuentra la Evaluación.

Para este estudio se aplica un instrumento individual de valoración respecto a cada uno de los tres problemas del examen parcial, ya enunciados. En base a la misma escala usada para los estudiantes, se solicita a los docentes que expongan el grado de complejidad global que cada problema le presenta al estudiante, según su estimación.

Además, que detallen la razón de su valoración global, distinguiendo los siguientes aspectos:

1. Adecuación de los componentes: el alumno debe modificar componentes para adecuarlos al problema (ejemplo: ordenamiento sobre vectores correspondientes).
2. Complejidad de la formulación: el alumno debe distinguir aspectos principales de accesorios que integran la formulación del problema
3. Complejidad de los componentes: el alumno debe aplicar componentes que clásicamente se consideran más complejos que otros (ejemplo: la búsqueda binaria presenta mayor complejidad que la búsqueda secuencial).
4. Contextualización de la formulación: el alumno debe situarse en una situación problemática real, distinguiendo de la formulación los aspectos principales que son esenciales para el algoritmo.
5. Estrategias múltiples de solución: el alumno debe elaborar un plan de resolución, evaluando diferentes estrategias y seleccionando la de menor complejidad algorítmica.
6. Estructuras de control: el alumno debe decidir aspectos relativos a secuencia, alternativas y ciclos, adoptando condicionados o incondicionados según corresponda.
7. Estructuras de datos: el alumno debe analizar los tipos de datos que correspondan, en el marco de la estrategia de su plan de resolución. Debe decidir sobre la conveniencia de utilizar datos simples o estructurados.
8. Generalización de la solución: el alumno debe diseñar un algoritmo que cubra casos de prueba normales y extremos.
9. Naturaleza del problema: el alumno debe identificar el o los componentes esenciales del problema (se trata de ordenar y luego buscar datos en una estructura).
10. Objetivo del problema: el alumno reconoce los datos de salida y la forma en que deben presentarse.

4. RESULTADOS

4.1. Valoración efectuada por la población estudiantil

Los 118 alumnos de la muestra provienen de una población de 306 estudiantes, los cuales registran inscripción en la cátedra. 98 alumnos (32%) no asisten a la instancia de

la primera evaluación parcial, aplicada al mes de cursado de la materia. Este porcentaje de abandono constituye un número histórico en esa etapa del cursado. Del 68% restante, 56 aprueban el primer parcial, constituyendo un porcentaje del 27%. A continuación se presentan las estadísticas más relevantes del estudio, respecto de la valoración de complejidad sobre los problemas de la evaluación.

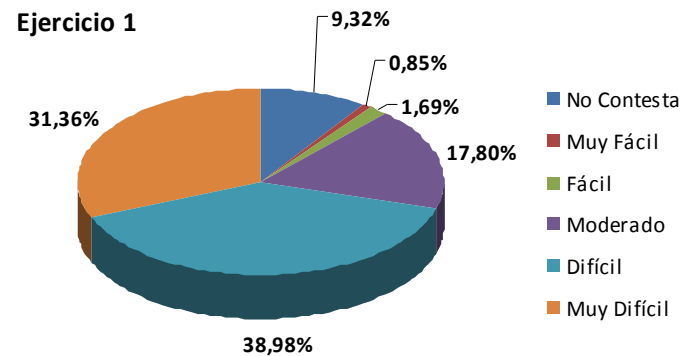


Gráfico 1. Valoración de estudiantes. Ejercicio 1

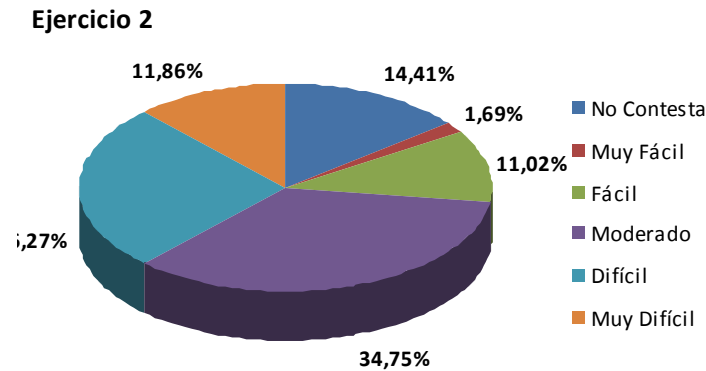


Gráfico 2. Valoración de estudiantes. Ejercicio 2

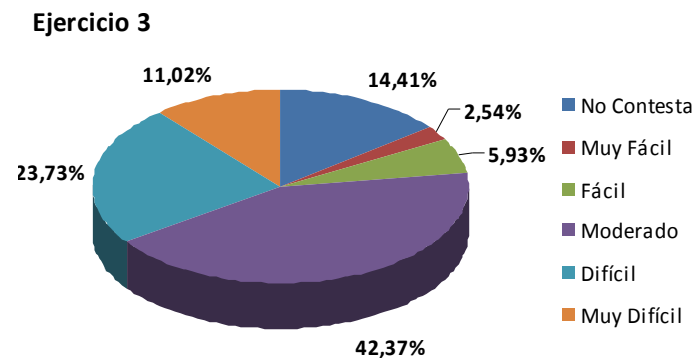


Gráfico 3. Valoración de estudiantes. Ejercicio 3

Reuniendo las categorías de Difícil y Muy difícil resulta, para cada ejercicio, lo que se muestra en la Tabla 1.

Valoración	Ejercicio 1	Ejercicio 2	Ejercicio 3
Difícil	46	31	28
Muy Difícil	37	14	13
Total	83	45	41
Porcentaje sobre respuestas válidas	78%	45%	41%

Tabla 1. Complejidad alta discriminada por ejercicio

Interesa entonces, en orden de complejidad, analizar en profundidad el problema 1. En la Tabla 2 se presenta el mismo estadístico para estudiantes aprobados.

El 33% de los estudiantes aprueban la evaluación, por obtención de una nota igual o mayor a 60 sobre 100 puntos. Este porcentaje constituye un valor histórico para el parcial que evalúa los contenidos descritos.

Valoración	Ejercicio 1	Ejercicio 2	Ejercicio 3
Difícil	17	6	8
Muy Difícil	14	1	2
Total	31	7	10
Porcentaje sobre respuestas válidas	79%	18%	28%

Tabla 2. Complejidad alta discriminada por ejercicio. Alumnos aprobados

Para el ejercicio 1, puede observarse la misma tendencia entre el colectivo de alumnos y los estudiantes aprobados. Claramente, el problema 1 es el que se reconoce como de mayor complejidad. Uno de los objetivos del estudio es analizar una posible correlación entre la valoración de los estudiantes y los docentes. Por tanto, se presentan a continuación los resultados arrojados por la aplicación del instrumento al interior de la cátedra.

4.2. Valoración efectuada por la población docente

Del análisis previo a la aplicación de la evaluación, el cuerpo docente que elabora y revisa, la considera equilibrada. En la prueba se distribuye equitativamente el puntaje entre los tres ejercicios, sin priorizar unos contenidos sobre otros.

Sólo los problemas 1 y 3 son consignados como difíciles, o incluso muy difícil (problema 3) para 1 de los 11 docentes (Auxiliar Estudiantil). El problema 2 es considerado moderado o fácil.

Los problemas 1 y 3 integran un 40% de docentes que lo valoran como difícil, con un 50% que lo valoran moderado.

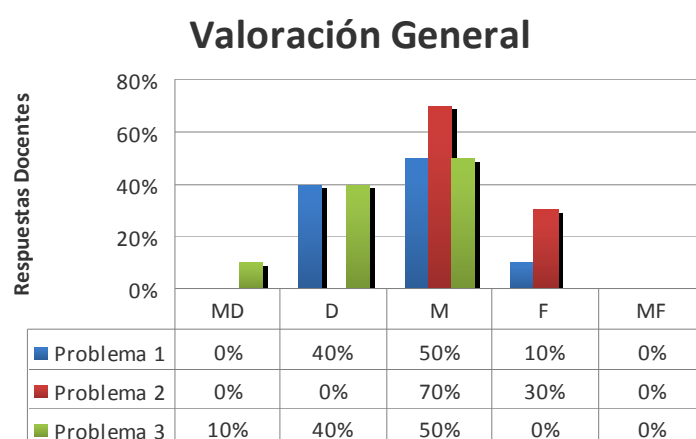


Gráfico 4. Valoración general de docentes

5. INTERPRETACIÓN DE RESULTADOS

Reuniendo las valoraciones de alumnos y docentes se reconoce que el problema 1 es el de mayor complejidad. Esto en razón de haber sido categorizado así por el 78% de los estudiantes, que lo consideran de difícil o muy difícil resolución y el 90% de los docentes, que lo consideran de difícil o moderada resolución.

Se presenta el estudio detallado de los criterios 2, 4, 5 y 9 que, en la valoración de los docentes, explican el alto grado de complejidad del problema 1.

Criterio 2. Complejidad de la formulación: el alumno debe distinguir aspectos principales, tales como contenido y tamaño de las estructuras de datos de entrada, de aquellos elementos accesorios del problema. Es usual que el estudiante inicial invierta un tiempo excesivo en el análisis de aspectos secundarios que pueden postergarse para cuando se tenga definida una estrategia de solución.

Criterio 4. Contextualización de la formulación: el alumno debe situarse en una situación problemática real, esto es, los elementos de la estructura que debe procesar no son “números”, son “días”, son “meses”. Este primer esfuerzo de abstracción le permitiría reconocer el problema como perteneciente a una familia de problemas cuya solución ya es conocida, practicada. Sin embargo, la transferencia de los aprendizajes a situaciones nuevas, en el marco de problemas reales que se apartan de las formulaciones académicas clásicas, son un verdadero escollo a superar, en la primera etapa formativa del razonamiento algorítmico.

Criterio 5. Estrategias múltiples de solución: es en este criterio en donde se observa una gran diferencia de dificultad con respecto a los otros problemas. El alumno debe elaborar un plan de resolución, que incluye un contraste entre diferentes estrategias. Es usual que el estudiante inicial priorice una estrategia inadecuada, sólo por el hecho que le son conocidos los pasos para implementarla. La menor complejidad algorítmica, no siempre es la aplicación de los algoritmos clásicos. Naturalmente, llegar a esta conclusión demanda un mayor nivel de experiencia.

Criterio 9. Naturaleza del problema: nuevamente, como en el criterio 5 y muy asociado a éste, existe una gran diferencia de complejidad de este problema sobre los otros dos.

Cada estrategia bajo análisis requiere la identificación de los componentes esenciales para su aplicación. Para concretar esta identificación, los estudiantes iniciales se esfuerzan por catalogar cada problema, ¿es de ordenar?, ¿de buscar?, ¿de eliminar o insertar datos? Una vez lograda esta catalogación –o al menos una aproximación a ella– adoptan la estrategia (criterio 5), aunque sin un cuidadoso análisis de los componentes esenciales. Esto se observa con mucha frecuencia, cuando presentan algoritmos incompletos o esbozos de algoritmos, en los que sólo puede apreciarse el “plan” de solución, no concretada por ausencia de los componentes.

6. CONCLUSIONES

La abstracción como proceso es esencial en la resolución de problemas a través del diseño algorítmico. Atendiendo a esta caracterización de la Programación, el Proyecto propone investigar procesos cognitivos y constructivos, que expliquen el modo en el que los estudiantes alcanzan el nivel de abstracción requerido y lo utilizan con el fin de desarrollar algoritmos y programas. A partir de allí, diseñar y aplicar configuraciones didácticas, en el marco de teorías de aprendizaje social.

Se reconoce que las actuales prácticas docentes están centradas en la reproducción memorística de conceptos y técnicas de la Programación, en un contexto artificial y academicista en el que el estudiante no encuentra satisfacción con la actividad de programar. “La programación de computadoras debe ser vista principalmente como una actividad de resolución de problemas que exige, por un lado, una forma metódica o sistemática de resolver problemas y por otro el diseño de un algoritmo” [1].

La cuestión no resuelta a partir de estas indagaciones, es el grado de convergencia (o peor, divergencia) entre los niveles de complejidad que estudiantes y docentes asignan a los mismos problemas. Siguiendo a Olmos [1], aunque se aplique una forma metódica de resolver problemas, es claro que existen rupturas manifestadas en el diseño algorítmico. Por tanto, es necesario que la investigación profundice el estudio de cada uno de los elementos, junto a su ponderación, presente en la metodología de análisis del problema, que contribuyen a la correspondiente solución algorítmica. Esto no es posible sin apelar a estrategias metacognitivas de los estudiantes, que guiados desde el aprendizaje reflexivo, manifiesten sus procesos de resolución de problemas computacionales.

7. REFERENCIAS

[1] Olmos, K., Morales, C., Rojas, T., Fernández, L. (2013). *Objetos de Aprendizaje Enfocados a la Resolución de Problemas para Facilitar la Enseñanza de la Programación*.

http://www.academia.edu/1303813/Objetos_de_Aprendizaje_Enfocados_a_la_Resolucion_de_Problemas_para_Facilitar_la_Ensenanza_de_la_Programacion

8. BIBLIOGRAFÍA

Castorina, J. A. y Carretero, M. (2012). *Desarrollo Cognitivo y Educación I*. Editorial Paidós. I.S.B.N: 9789501261622.

García de Fanelli, A. M. *Acceso, abandono y graduación en la educación superior argentina*. En <http://www.siteal.iipe-oei.org>.

González de Rivera Fuentes, M. y Paredes Velasco, M. (2008). *Aprendizaje con programación Colaborativa*. En Serie de Informes Técnicos DLSII-URJC Número 2008-02. ISSN 1988-8074.

Jiménez Rey, E., Rodríguez, D., Britos, P., García Martínez, R. (2008). *Identificación de problemas de aprendizaje de Programación con explotación de información*. En XIV Congreso Argentino de Ciencias de la Computación.

Saltalamacchia, H. *Aportes para una investigación socialmente útil. Sujetos, Teoría y Complejidad* (2005). Tomo I. Ed. El Artesano. Buenos Aires.

Sánchez, F., Quirós, M., Reverón, C., Rodríguez, A. (2006). *Equidad Social en el Acceso y Permanencia en la Universidad Pública. Determinantes y Factores Asociados*. Documento en línea.