



**CONGRESO
IBEROAMERICANO**
DE CIENCIA, TECNOLOGÍA,
INNOVACIÓN Y EDUCACIÓN

BUENOS AIRES, ARGENTINA
12, 13 Y 14 DE NOVIEMBRE 2014

**CONGRESSO
IBERO-AMERICANO**
DE CIÊNCIA, TECNOLOGIA,
INOVAÇÃO E EDUCAÇÃO

BUENOS AIRES, ARGENTINA
12, 13 Y 14 DE NOVIEMBRE 2014

LA COMPLEJIDAD DEL DISEÑO ALGORÍTMICO DESDE LA PERSPECTIVA DOCENTE Y DEL ESTUDIANTE UNIVERSITARIO INICIAL

MAC GAUL, M; MASSÉ, M; VARGAS, C; DEL OLMO, P; LÓPEZ, M; FERNÁNDEZ, E;
RIVERA, A

LA COMPLEJIDAD DEL DISEÑO ALGORÍTMICO DESDE LA PERSPECTIVA DOCENTE Y DEL ESTUDIANTE UNIVERSITARIO INICIAL

Mac Gaul, Marcia; Massé, María Laura; Vargas, Claudio; del Olmo, Paola; López, Marcela; Fernández, Eduardo; Rivera, Ariel

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

mmacgaul@cidia.unsa.edu.ar - mlmassep@cidia.unsa.edu.ar cvargas@unsa.edu.ar -
pdelolmo@unsa.edu.ar - mfflopez@unsa.edu.ar effer@cidia.unsa.edu.ar -
riverae8080@yahoo.com.ar

RESUMEN

Si bien en los últimos años y fruto de políticas educativas propiciatorias, en las universidades públicas hubo una fuerte expansión de la matrícula, este no es un hecho recurrente en las carreras científico-tecnológicas. En particular, las carreras de Informática muestran una tasa decreciente de inscripción y alto grado de abandono, sobre todo en primer año.

El aprendizaje de la Programación es un proceso complejo para los estudiantes universitarios iniciales, ya que requiere de la apropiación de habilidades específicas que en muchos casos, no han sido abordadas 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, construidos 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.

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. Se centra en indagar 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.

La indagación realizada permite analizar una población de 118 estudiantes y 11 docentes. Los alumnos fueron indagados en la resolución de tres problemas computacionales adecuados a estudiantes iniciales de programación. El estudio se focaliza en dos aspectos, el primero corresponde a la relación entre la solución planteada y el grado de dificultad que el alumno asigna a cada problema; el segundo toma la palabra de los docentes de la cátedra, para analizar el sentido que otorga a la evaluación y particularmente a la complejidad que creen que cada uno de estos problemas presenta.

A partir de las conclusiones se elaboran orientaciones de diseño de los materiales educativos aplicados a la enseñanza del diseño algorítmico.

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 expansión de matrícula es una característica de la última década en las universidades públicas. La inclusión educativa fue tema central en la agenda de las políticas educativas. Sin embargo, en los últimos años se comenzó a observar que en las carreras científico-tecnológicas, crecen los índices de abandono, sobre todo en primer año y la prolongación de las carreras en el tiempo. Los estudiantes ingresan y no logran “permanecer” ni realizar sus estudios “en los tiempos estipulados” desde la estructura curricular de los Planes de Estudios vigentes.

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, construidos 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.

La cátedra de Elementos de Programación está integrada por dos profesores que imparten clases teóricas y nueve auxiliares docentes dispuestos en ocho comisiones de clases prácticas. Posee una carga horaria semanal de diez horas, distribuidas en cuatro de teoría y seis de práctica. Los contenidos pueden distinguirse en tres grandes ejes:

- Conceptos iniciales de la Programación, con énfasis en el diseño de algoritmos (aproximadamente 90 de las 150 horas de cursado).
- Elementos de computación básicos que se asientan sobre formalizaciones de la Matemática Aplicada: Sistemas de Numeración y Álgebra de Boole.
- Contenidos complementarios e introductorios orientados a la alfabetización informática.

Los conceptos iniciales 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 cursado, en el primer cuatrimestre del año 2014.

2. METODOLOGÍA

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

La indagación se centra en dos aspectos, 1) el grado de dificultad que el alumno asigna a cada problema y su posible relación entre esta valoración y la solución algorítmica que realiza; 2) Desde la palabra de los docentes de la cátedra, el sentido que se otorga a la evaluación y particularmente al grado de complejidad que creen que cada uno de estos problemas presenta para sus estudiantes.

2.1. 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 los tres problemas que se muestran a continuación, cuya solución debe presentarse usando Diagramas N-S¹, como herramienta de especificación algorítmica.

Problema 1: Se poseen N fechas en dos vectores correspondientes D y M . El vector D contiene números naturales correspondientes a los días y el vector M contiene números naturales correspondientes a los meses. Se desea mostrar un reporte organizado por mes, desde 1 a 12, que muestre las fechas válidas y las fechas inválidas, así como la cantidad de cada una de ellas.

Se recuerda que los meses de 31 días son: 1 (enero), 3, 5, 7, 8, 10, 12. Los meses de 30 días son: 4, 6, 9, 11 y el único mes de 28 días es el mes 2. En este problema no se considera año bisiesto.

Ejemplo: si $N = 15$ y los vectores son:

$D = (12, 4, 7, 9, 19, 31, 30, 5, 3, 2, 12, 33, 21, 5, 15)$

$M = (12, 11, 9, 11, 1, 11, 4, 8, 12, 3, 1, 2, 2, 11, 10)$

La salida es:

Mes 1:

Válidas: 19/1; 12/1. Cantidad de válidas: 2

Inválidas: --- Cantidad de inválidas: 0

Mes 2:

Válidas: 21/2. Cantidad de válidas: 1

Inválidas: 33/2. Cantidad de inválidas: 1

.....

¹ Nassi Shneiderman Diagrams.

Mes 11:

Válidas: 4/11; 9/11; 5/11. Cantidad de válidas: 3

Inválidas: 31/11. Cantidad de inválidas: 1

Mes 12:

Válidas: 12/12; 3/12. Cantidad de válidas: 2

Inválidas: --- Cantidad de inválidas: 0

Problema 2: Dado un vector V de N elementos reales ordenados de menor a mayor y sin elementos repetidos, se pide modificarlo para que todos los elementos sean números enteros, según el siguiente criterio:

- Si un elemento tiene una parte fraccionaria menor o igual a 0.5 se debe convertir el elemento al menor número entero más próximo a él.
- Si un elemento tiene una parte fraccionaria mayor a 0.5 se debe convertir el elemento al mayor número entero más próximo a él.

Finalmente, calcular la parte entera del promedio (PEP) entre los nuevos elementos de V e informar si está o no en el vector. En caso de estar en V , informar en que posición.

Ejemplo: si $N = 10$ y $V = (5.62, 10.3, 10.5, 21, 31.8, 33.99, 34.12, 34.22, 40, 48.5)$

La Salida es: $V = (6, 10, 10, 21, 32, 34, 34, 34, 40, 48)$. $PEP = 26$. No está en V .

Problema 3: Se posee una matriz A de M filas y N columnas cuyos elementos son números naturales, en la que todas las filas, excepto una, están ordenadas de menor a mayor. Se solicita eliminar de la tabla A la fila no ordenada y con esos elementos formar un vector V y mostrarlo ordenado ascendentemente.

Ejemplo: si $M = 4$, $N = 3$ y la tabla es:

$$A = \begin{vmatrix} 1 & 7 & 12 \\ 10 & 10 & 12 \\ 5 & 4 & 9 \\ 1 & 21 & 21 \end{vmatrix} \quad \text{La Salida es:} \quad A = \begin{vmatrix} 1 & 7 & 12 \\ 10 & 10 & 12 \\ 1 & 21 & 21 \end{vmatrix} \quad V = (4, 5, 9)$$

2.2. Población estudiantil

La población estudiantil que ingresa transita con problemáticas que impide su permanencia en los estudios universitarios. Las mismas tienen que ver con baches teóricos y metacognitivos de “arrastre” desde el Nivel Secundario. El aprendizaje constituye un proceso por el cual el sujeto pone de manifiesto los diferentes momentos por los que atraviesa, en su aproximación a explicaciones cada vez más adecuadas. Son estos procesos los que se necesita indagar, para aproximarnos a las problemáticas y dificultades de tipo cognitivo y reconfigurar el espacio didáctico.

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.

2.3. 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 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.

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.

Las evaluaciones parciales de Elementos de Programación son elaboradas por un Profesor, a partir de su propia contribución y las propuestas de problemas efectuadas por un par de Jefes de Práctica. Una vez elaborada esta primera versión, el otro Profesor revisa y –eventualmente- sugiere modificaciones en los problemas, de tal forma que el instrumento de evaluación resulte con una complejidad equilibrada.

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.

3. RESULTADOS

3.1. 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.

Noventa y ocho 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%.

La cátedra aplica esta primera evaluación como una estrategia retentiva, ya que la división del extenso contenido asociado a la Diagramación Algorítmica, contribuye a la reducción de la complejidad del primer parcial.

A partir del cumplimiento de actividades evaluativas previas, dispuestas en el entorno virtual de la cátedra, se elabora la nómina de los 118 individuos habilitados a presentarse en la segunda evaluación parcial.

Esta modalidad de *coloquios virtuales*, usualmente de resolución a través de alternativas múltiples de respuesta, es otra estrategia retentiva, debido a que el estudiante puede resolver los coloquios sin límite de tiempo, desde cualquier computadora, consultando los textos e incluso trabajando en grupo. Su principal finalidad es generar una actitud autónoma y responsable frente a la agenda universitaria. Para los alumnos es una instancia de auto evaluación, pauta al mismo ritmo en que se desarrollan los contenidos. Por otra parte, la obligatoriedad de los coloquios establece la necesidad de conectarse al aula virtual de la cátedra, estimulando el interés por participar en los foros, visitar otros portales de interés

académico y mantenerse informado sobre los aspectos organizativos de una cátedra de acceso masivo.

A continuación se presentan las estadísticas más relevantes del estudio, respecto de la valoración de complejidad sobre los problemas de la evaluación.

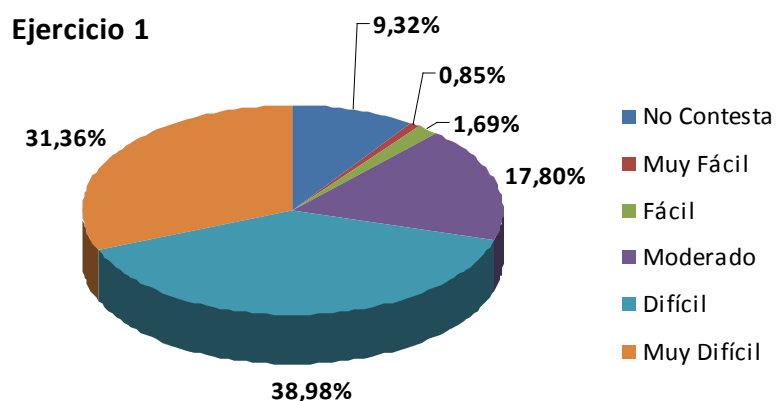


Gráfico 1. Valoración de estudiantes. Ejercicio 1

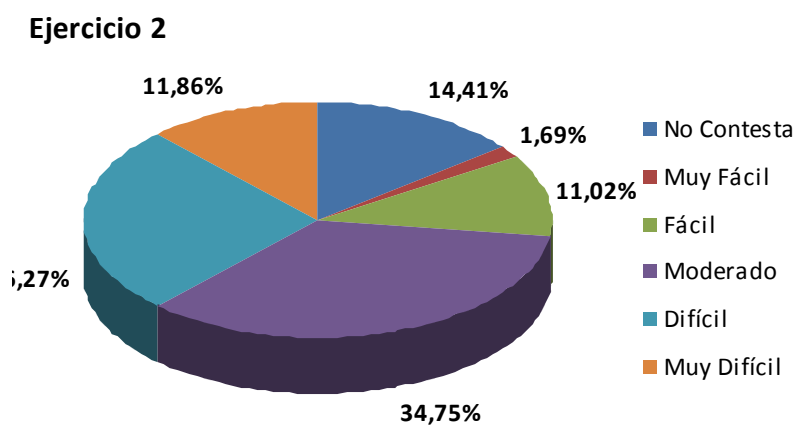


Gráfico 2. Valoración de estudiantes. Ejercicio 2

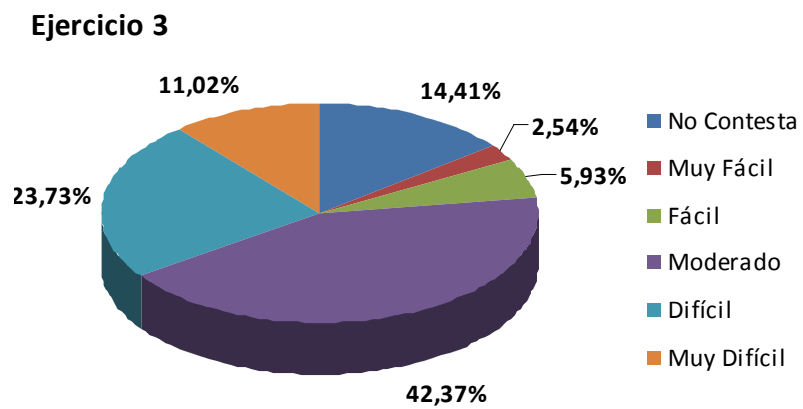


Gráfico 3. Valoración de estudiantes. Ejercicio 3

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.

3.2. 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.

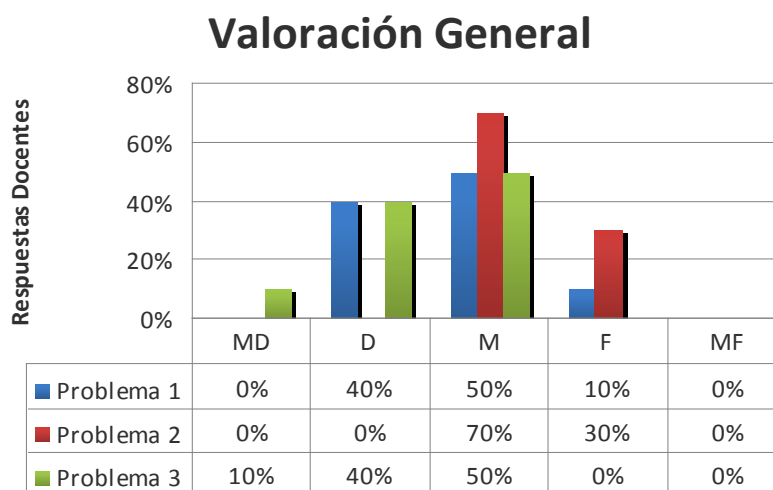


Gráfico 4. Valoración general de docentes

4. 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.

Comparativo. Criterio 2

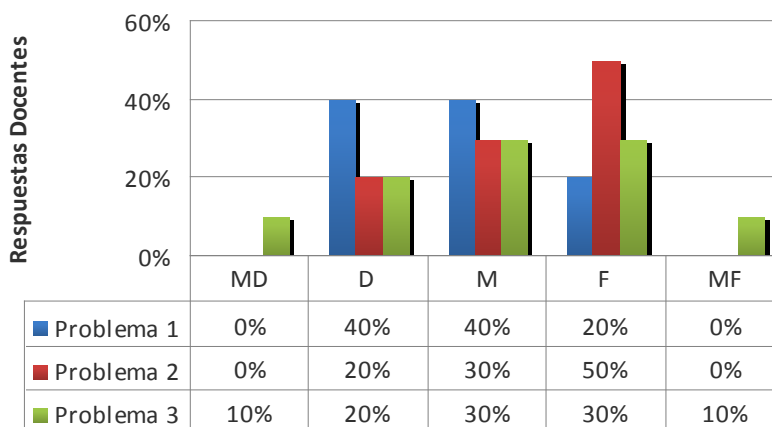


Gráfico 5. Criterio 2

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, tal como el criterio para definir la cantidad de días que posee un determinado mes. 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.

Comparativo. Criterio 4

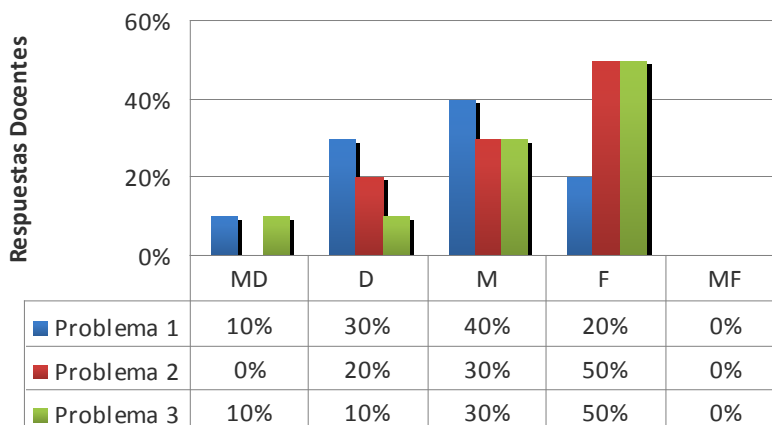


Gráfico 6. Criterio 4

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.

Comparativo. Criterio 5

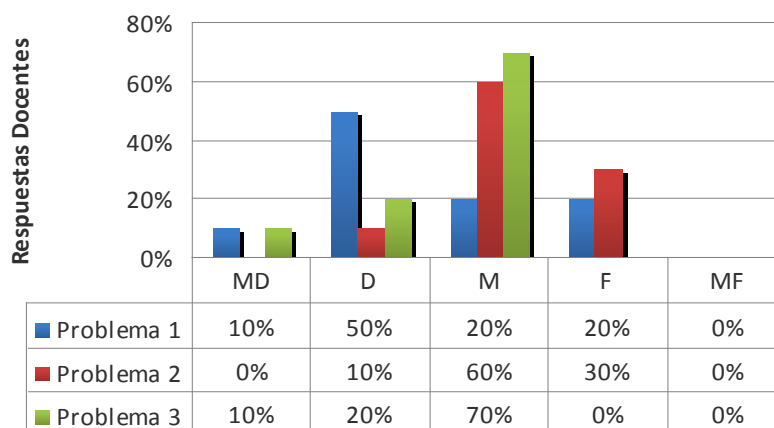


Gráfico 7. Criterio 5

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. Por ejemplo, se observa que numerosos alumnos ordenan los datos de día y mes, según su posición en el calendario, pues ordenar es una tarea algorítmica sencilla, sumamente practicada. Son menos los que advierten que el ordenamiento más natural es el que surge de aplicar un criterio de recorrido de los datos, desde el mes 1, 2 y así sucesivamente, hasta el mes 12. La menor complejidad algorítmica, no siempre es la aplicación de los algoritmos clásicos. Naturalmente, llegar a esta conclusión demanda un mayor nivel de experiencia.

Comparativo. Criterio 9

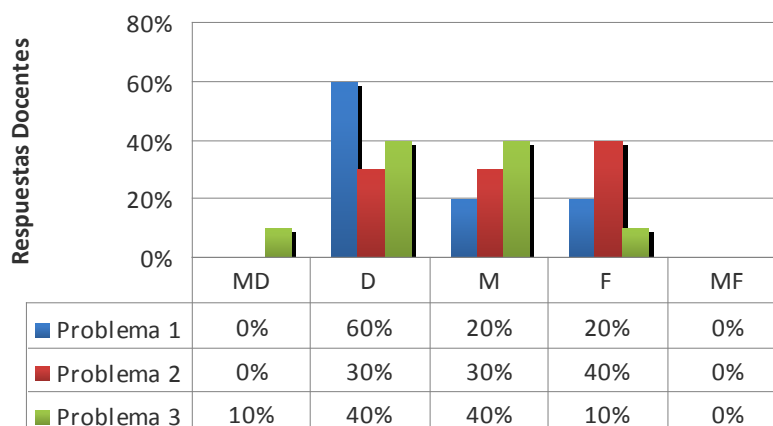


Gráfico 8. Criterio 9

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.

5. FUTURAS LÍNEAS DE ACCIÓN

La investigación continuará aplicando entrevistas en profundidad a una muestra de estudiantes. Se indagará acerca de las posibles rupturas al momento de ser sometidos al diseño algorítmico de problemas como los presentados en este estudio.

Debido a que este proyecto se orienta hacia la elaboración y adecuación de medios educativos tecnológicos, interesa conocer hasta qué punto les favorece u obstaculiza el diseño algorítmico, usando la aplicación *Diagrama*². Algunos servicios de esta

² Intérprete gráfico de Diagramas N-S elaborado en el marco de una investigación previa e implementado en la asignatura Elementos de Programación.

aplicación, como la que permite escoger componentes de una galería de procesos (ordenar, buscar, etc.) ¿liberan al alumno de su acabada comprensión? o –por el contrario- le permite concentrarse en los aspectos esenciales de la elaboración de un plan de resolución, postergando los aspectos de su implementación. La posibilidad de ejecutar una prueba automatizada con *Diagramar*, ¿dificulta que el estudiante efectúe una prueba manual de sus algoritmos en una instancia de evaluación escrita, sin computadora? Si en las clases se estimula el uso de *Diagramar* para el diseño algorítmico, ¿la evaluación debería desarrollarse en el mismo entorno?, ¿el cuerpo docente está preparado para esa posibilidad? Éstas y otras preguntas son objeto de estudio.

6. CONCLUSIONES

La abstracción 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, 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. Esta investigación se plantea, ¿podría reemplazarse por un trabajo entre pares y expertos conformando una red apoyada por TIC, en un contexto de producción de software más cercano al del ámbito profesional?, ¿podría revertirse el prejuicio que la Programación es una actividad antisocial y solitaria? “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].

Se plantea también la contribución de las TIC en el sostenimiento de estrategias de intervención, ya que brindan la posibilidad de hacer un seguimiento sistemático de los procesos de aprendizaje e intervenciones oportunas, una mayor autonomía del estudiante, una mayor integración y cohesión de grupos de programación durante la elaboración y socialización de las producciones.

Desde la perspectiva docente, de investigaciones anteriores se reconoce también, que el tránsito que recorre un docente con enfoque tradicional hacia un docente

preocupado en la construcción y socialización del conocimiento de sus alumnos, debe actualizarse y adecuarse, de manera que afiance las nuevas prácticas con el uso sostenido de la tecnología, reconociendo además, que la mayoría de los profesores que se desempeñan en las asignaturas de Informática no poseen formación docente. En este sentido, las cátedras de primer año vienen introduciendo cambios respecto a los marcos teóricos de enseñanza y la incorporación de TICs para el sostenimiento de un aprendizaje constructivista y social. A pesar de que estas intervenciones pudieron ser evaluadas como exitosas en su oportunidad, los docentes y alumnos dejaron de utilizar muchos de los recursos tecnológicos del contexto de enseñanza, volviendo a modos tradicionales, con clases expositivas, prácticas guiadas y alumnos pasivos.

Por tanto, el desafío en este proyecto consiste en explorar los modos en que aprenden y se comunican los jóvenes y las posibilidades didácticas de los medios tecnológicos que usan, con la finalidad que los docentes puedan internalizar estos modos y a partir de ello replantear sus prácticas, a fin de favorecer recorridos menos complejos.

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

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 DLSI1-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.