

## Hacia una propuesta didáctica para la enseñanza de la programación

Gustavo ASTUDILLO<sup>1</sup>Silvia BAST<sup>1</sup>Yamila MINETTI<sup>1</sup>

<sup>1</sup> *Grupo de Investigación y Desarrollo en Informática Educativa en (GrIDIE)-  
FCEyN/UNLPam*

astudillo@exactas.unlpam.edu.ar, silviabast@exactas.unlpam.edu.ar,  
minettiyamila@humanas.unlpam.edu.ar

### Resumen

Hay una tendencia, a nivel global, que se enfoca en la generación de propuestas para la enseñanza de la programación de computadoras. Desde el GrIDIE<sup>1</sup> en los dos últimos años se viene desarrollando una propuesta didáctica para el aprendizaje de nociones básicas de programación. La misma fue evaluada desde el equipo docente, concluyendo que al abordar lo conceptual, había dejado de lado un aspecto indisoluble en la creación de programas: la resolución de problemas.

Este artículo presenta una propuesta para enseñar a resolver problemas en el contexto de la programación. La misma se nutre de los aportes en didáctica de la programación de la Fundación Sadosky, que se combinan con la estrategia de resolución de problemas que presenta Thomson.

La propuesta didáctica resultante, hace uso de la didáctica por indagación, la abstracción y la división en subtarear y fue implementada por primera vez en 2020. Los resultados, aunque preliminares, dan cuenta del potencial de esta.

**Palabras Clave:** Programación, enseñanza, resolución de problemas, indagación, abstracción

### Introducción

A nivel mundial existe una tendencia que aboga por la inclusión de la programación de computadoras en instancias cada vez más

tempranas de la educación formal. Esto, según [1] se puede atribuir a tres razones: (i) preparar a la fuerza laboral del futuro con conocimientos de programación, (ii) un ideal empresarial de estudiantes que aprenden a ser "productores" de innovación y (iii) situar al aprendizaje de la programación como competencias del siglo XXI.

Argentina no es la excepción y una muestra de esto es que, desde el Gobierno, se vienen impulsando distintas normas que apoyan el proceso [2]–[5]. Sin duda la más importante ha sido en el año 2018, la aprobación de los Núcleos de Aprendizaje Prioritarios (NAP) para educación digital, programación y robótica, con el fin de “facilitar la integración del acceso y dominio de las tecnologías de la información y la comunicación en los contenidos curriculares indispensables para la inclusión en la sociedad digital” [6]. Esto último implica la incorporación de la temática al currículum de cada provincia de la Argentina.

A lo anterior se debe sumar las iniciativas de la Fundación Sadosky<sup>2</sup>, que promueve “el estudio de la programación en las escuelas argentinas y [...] para ello lleva adelante diversos programas, entre estos, Vocaciones en TIC y Program.ar.” [7]. Particularmente, “Program.AR es una iniciativa que trabaja para que el aprendizaje significativo de Computación esté presente en todas las escuelas argentinas.” [8].

<sup>1</sup> Grupo de Investigación y Desarrollo en Innovación Educativa. FCEyN-UNLPam.

<sup>2</sup> Accesible en:  
<http://www.fundacionsadosky.org.ar>

También debe mencionarse la ingente cantidad de propuestas didácticas con el propósito de que estudiantes de distintos niveles educativos aprendan a programar [9], [10]. Sin embargo, como afirma [11] “inicialmente la metodología aplicada [para enseñar programación] sigue el modelo de la escuela pedagógica tradicional, un modelo por imitación, en donde el instructor propone un problema y desarrolla e implementa su solución, esperando que el estudiante lleve este desarrollo a su propio contexto” (p. 58), sumado a esto, las/los estudiantes siguen aprendiendo herramientas de ofimática (propuesta instrumental) y en muy poca proporción conceptos de programación [12].

Desde la cátedra Introducción a la Computación (Facultad de Ciencias Exactas y Naturales-UNLPam), se vienen implementando y evaluando, distintas estrategias para mejorar el acercamiento de las/los estudiantes a las nociones básicas de programación. Desde 2010, en conjunto con el Grupo de investigación GrIDIE, se comenzó a desarrollar una propuesta didáctica para un Taller de Introducción a la Programación (TIP) [13]. El TIP se enfoca, fuertemente, en los conceptos básicos de la programación (secuencia, estructuras de control, expresiones, datos simples y variables), dejando en un segundo plano la resolución de problemas. Por este motivo se plantea la incorporación de estrategias de resolución de problemas desde el inicio de la cursada de Introducción a la Computación.

El artículo se organiza como sigue: inicia con el marco teórico y conceptual que sustenta la propuesta didáctica. Continúa con una descripción de la propuesta, los resultados y la discusión. Finalmente, se presentan las conclusiones y trabajos futuros.

## Marco Teórico

### La enseñanza

Si bien “enseñar” es un término muy utilizado y que, en principio, es conocido por quienes ejercen la docencia, conviene echar luz sobre el mismo antes de continuar.

Dicen [14] “puede definirse a la enseñanza como un intento de alguien de transmitir cierto contenido a otra persona” (p. 126).

Entonces ¿Qué pretendemos hacer cuando diseñamos una propuesta de enseñanza? En principio debemos tener en cuenta que “la enseñanza es siempre una forma de intervención destinada a mediar en la relación entre un aprendiz y un contenido a aprender [y] abarca indistintamente tanto a los esfuerzos infructuosos realizados para que alguien aprenda algo, como a las ocasiones en las que ello efectivamente sucede, [por lo que] entre los procesos de enseñanza y aprendizaje no hay una relación de tipo causal que permita asumir que lo primero conduce necesariamente a lo segundo.” [14].

Otro aspecto importante es tener en cuenta cómo, cuándo, dónde, para quiénes se llevará adelante la propuesta de enseñanza. Debe diseñarse en un espacio, en un tiempo, para un grupo de estudiantes, entre otros condicionantes, que darán a la misma su carácter contextual y situado. Dicen [15] “la buena enseñanza es aquella con intencionalidades definidas y explícitas, que promueve la interacción entre los alumnos y los docentes, y los alumnos entre sí, y que transcurre en un espacio, tiempo y en un contexto socioeconómico determinado.” (p. 117)

Dicho esto, la propuesta aquí descrita tiene la pretensión de mediar entre los conocimientos previos de las/los estudiantes y la construcción de los nuevos saberes implicados en la resolución de problemas. Así, brindar a las/los

estudiantes la posibilidad de apropiarse de una estrategia que puedan transferir a otros contextos.

### La resolución de problemas

Dice [16] “Formar a un niño o a un joven en la siempre móvil sociedad contemporánea requiere ofrecerle herramientas que le permitan comprender la realidad compleja en la que vive. Simultáneamente se le ha de ayudar a adquirir las competencias necesarias para desenvolverse en esta realidad” (p. 1) Esto incluye, claramente, enseñarles a resolver problemas.

Según [14] “enseñar es plantear problemas a partir de los cuales sea posible reelaborar los contenidos [...] y es también proveer toda la información necesaria para que los [estudiantes] puedan avanzar en la reconstrucción de esos contenidos.” (p. 130).

Pero ¿Qué es un problema? De acuerdo a [17] para ser un problema no debemos disponer de “procedimientos de tipo automático que nos permitan solucionarlo de forma más o menos inmediata, sino que requieren de algún modo un proceso de reflexión o toma de decisiones sobre la secuencia de pasos a seguir.” (p. 17).

La resolución de este tipo de situaciones problema implica, entonces, utilizar de modo estratégico técnicas ya conocidas [18]. En este sentido dice [19] “Toda vez que se busca solucionar algún problema, es necesario contar primero con alguna idea de cómo encarar dicha solución [...]. Esto en programación se conoce como estrategia de solución” (p. 9).

Para algunos autores, la solución de un problema exige una comprensión de la tarea, diseñar un plan, la ejecución de este y, por último, un análisis que nos lleve a determinar si se ha alcanzado una solución [17], [20]. En este sentido, Thomson [21] propone la siguiente secuencia para ayudar a sus estudiantes a programar: comprender el

problema; diseñar el programa; escribir el programa y finalmente mirar hacia atrás (o "reflexión").

Para la etapa inicial (comprender el problema o *understanding*) [21], [22] proponen analizar el enunciado, dividirlo en partes más pequeñas, de forma de comprender el ámbito del problema, determinar las entradas, las salidas y las especificaciones del problema. También, usan la pregunta *¿Y si? (What if?)* en busca de aspectos que no estén claros en el enunciado y para la generación de posibles casos de prueba [22]. En la segunda etapa (diseñar el programa o *design*), los autores se enfocan en la búsqueda de problemas similares ya resueltos por las/los estudiantes y el uso de los ejemplos hallados en la etapa anterior para probar el diseño. La tercera etapa (escribir el programa o *writing*), se trata de dar forma al diseño a través de un lenguaje de programación, los autores sugieren el reuso de código ya escrito. Finalmente, mirar hacia atrás (o *review or looking back*); en esta etapa, se vuelve hacia atrás y revisa el producto terminado, el “objetivo aquí es consolidar el proceso de aprendizaje y apreciar las lecciones aprendidas.” [22, p. 3].

### La importancia de la pregunta

Parte de esta propuesta de enseñanza se basa en la pregunta como estrategia didáctica. La resolución de problemas inicia con la comprensión del mismo a través de la lectura/relectura del enunciado, y luego aparece la pregunta disparadora: *¿Qué hay que hacer?*

Se cuestionan en [15], los docentes, ¿Tienen clara conciencia de qué se proponen cuando preguntan? En este caso, la pregunta pretende ser un disparador y activador de los conocimientos previos, y se busca que sea la primera pregunta que se haga el estudiante cuando tiene que resolver un problema.

De acuerdo con el nivel de pensamiento que intentan estimular, se puede distinguir entre cuatro tipos de preguntas [15]: (i) preguntas sencillas, que requieren de respuestas breves, casi siempre únicas, informaciones precisas, en general, no generarán un diálogo; (ii) preguntas de comprensión, se proponen estimular el procesamiento de información e involucra acciones como relacionar datos, clasificar, etc. para dar una respuesta; (iii) preguntas de orden cognitivo superior, la cuales exigen interpretar, predecir, evaluar críticamente, para construir la respuesta; (iv) y preguntas metacognitivas, que se proponen ayudar a los alumnos a reflexionar sobre su modo de aprender y de pensar.

### **Los organizadores gráficos**

Las imágenes tienen la función tanto de focalizar la atención de estudiantes, como de contribuir a su comprensión de un concepto o problema. Es así como, en esta propuesta, se hace uso de las imágenes de los formularios (aspecto de la ventana del programa que resolverá el problema planteado) como estrategia didáctica.

Estas imágenes, a las que se le agregan las distintas partes del plan (diseño del programa), hacen las veces de organizadores gráficos. “Los organizadores gráficos son formas de representar, a través de esquemas, los conceptos y las relaciones entre estos. Habitualmente, los conceptos se colocan en un recuadro u óvalo (nodo); y las relaciones entre ellos se representan mediante líneas que muestran la interconexión.” [15].

En este caso, la imagen del formulario hace las veces de nodo central y, de los distintos objetos que componen el mismo, se desprenden flechas a otros nodos que representan cada funcionalidad asociada al objeto. Esta se deberá programar para alcanzar la solución del problema.

Dicen [15] “los organizadores gráficos sirven para construir, comunicar y negociar significados” (p. 74) y en ese sentido la discusión sobre qué colocar en cada nodo (funcionalidad) no sólo avanza hacia la solución del problema, sino también hacia una mejor comprensión del mismo al poder expresar en palabras cuál será la acción/es que se espera realicen los objetos para resolver el problema y, si al completar el organizador se ha dado solución a este. También, una vez que se ejecuta el plan (programar), el organizador hace las veces de guía a la que se puede volver una y otra vez para ver cómo seguir. Asimismo, el organizador se puede actualizar en pos de una mejor solución.

### **La propuesta de la Fundación Sadosky**

Desde la Fundación Sadosky (FS) se diseñó una propuesta didáctica para la enseñanza de los conceptos básicos de programación. Esta hace uso del aprendizaje por indagación como estrategia didáctica.

El proceso de indagación debe partir de una experiencia real del estudiante. Este debe identificar el problema a partir de esa experiencia, inspeccionar los datos disponibles, formular la hipótesis de solución y comprobar esta por la acción [23].

La propuesta didáctica de la Fundación Sadosky trabaja con un conjunto de herramientas conceptuales [19]: (i) la noción de estrategia de solución (división en subtareas); (ii) la noción de que los programas deben poder comunicar tanto a máquinas como a personas (legibilidad); y (iii) la noción de algorítmica básica (secuencia/recorrido).

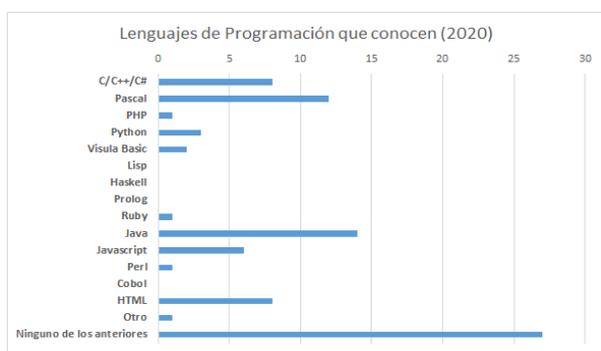
La estrategia de solución es expresada a través de la división en subtareas, la misma “representa a la forma de pensar composicionalmente y es una herramienta invaluable en el pensamiento de alto orden” [19, p. 9]. Dicha estrategia es representada a través del uso de procedimientos. Cada

solución a un problema (programa) inicia con un único procedimiento con un nombre que refleje la estrategia de solución (el problema como un todo, en nuestro caso la respuesta a la pregunta ¿Qué hay que hacer?), este a su vez se irá dividiendo en tantas subtarefas como sea necesario (nombrando cada una adecuadamente, iniciando con un verbo que describa la acción).

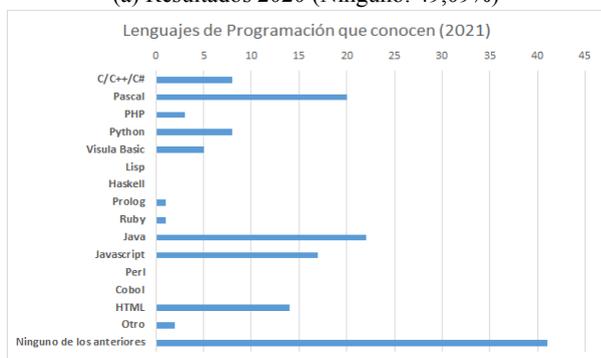
## La propuesta de enseñanza

### El contexto

Si bien la situación varía año a año, cada vez son más las/los estudiantes que llegan a Introducción a la Computación con conocimientos básicos de un lenguaje de programación (principalmente los inscriptos en el Profesorado en Computación, y en menor medida Profesorado/Licenciatura en Matemática (FCEyN-UNLPam)). Sin embargo, en general, no cuentan con experiencia en programación (ver Figura 1).



(a) Resultados 2020 (Ninguno: 49,09%)



(b) Resultados 2021 (Ninguno: 50%)

Figura 1. Encuesta inicial de la cátedra, pregunta “¿Conocés alguno de los siguientes lenguajes de programación?”

Los objetivos de aprendizaje planteados desde la asignatura son: (i) Comprender y aplicar una metodología de resolución de problemas en situaciones prácticas, (ii) Comprender y utilizar apropiadamente estructuras de control y tipos de datos en la resolución de un problema dado y (iii) Diseñar programas en un lenguaje de alto nivel haciendo uso de un entorno integrado de desarrollo (IDE). Por lo que, no sólo se busca que las/los estudiantes se apropien de las nociones básicas de la programación, sino también, que aprendan una metodología para resolver problemas.

Como afirman [14] “Las intenciones educativas se expresan, habitualmente, en las propuestas curriculares –especialmente en las formulaciones de objetivos, propósitos y contenidos a enseñar– que constituyen un importante marco de regulación de la tarea del profesor.” (p. 141). De aquí la importancia y el compromiso de diseñar y desarrollar una propuesta de enseñanza que incluya la resolución de problemas.

Con base en los resultados de los parciales, se pudo identificar que la sintaxis del lenguaje Pascal, y el uso del IDE Lazarus no son un obstáculo para la resolución de los problemas. Por lo que se decidió poner el foco en las estrategias de resolución de problemas.

Por otra parte, la FCEyN-UNLPam, firma un convenio con la Fundación Sadosky para desarrollar el curso La Programación y su Didáctica 1 y en este contexto, los integrantes del equipo docente de Introducción a la Computación fueron capacitados por la Fundación y llevaron adelante dicha propuesta didáctica en un curso durante 2020.

### Propuesta de enseñanza para la resolución de problemas

La presente propuesta de enseñanza para el aprendizaje de una estrategia de resolución de problemas toma como base las etapas propuestas por Thomson (ya abordadas en la

cátedra) y la metodología de la Fundación Sadosky (ya descrita en el marco teórico).

Como recurso didáctico, se cuenta con las Guías Prácticas (GP). Estas incluyen ejercicios y problemas que se presentan con una complejidad creciente. Las temáticas abordadas en las mismas son: Conociendo el ambiente Lazarus, Alternativa condicional, Repetitivas, Arreglos y matrices. Inician con una introducción a la temática (marco conceptual), luego se presentan problemas y ejercicios de forma intercalada. Los primeros buscan trabajar un saber y los segundos transferir los aprendizajes. Las GP cierran con un resumen de lo abordado.

Para cada clase se selecciona un problema de la GP y se discute la solución con las/los estudiantes, aquí es donde se pone en práctica la propuesta de enseñanza. Esta incluye:

**Comprender el problema.** Se presenta a las/los estudiantes el enunciado del problema y se pregunta *¿Qué hay que hacer?* (pregunta de orden cognitivo superior). Se espera aquí que puedan definir en una oración corta cuál es el problema a resolver.

Se registran las respuestas y se discute cuál sería la oración que mejor responde la pregunta *¿Qué hay que hacer?*

Se vuelve a colocar el enunciado, pero ahora, destacando en el mismo la parte donde se encuentra la frase que describe el problema. En este punto, se solicita que se desglose el problema en partes más pequeñas (división en sub-tareas). Se discuten las ideas propuestas y se define el conjunto de partes en que se dividirá el problema (pensar el diseño o el plan de solución).

Se analiza si se han resuelto problemas similares que pudieran aportar a la solución de este nuevo problema.

**Diseñar el programa.** Con base en el ejemplo del formulario (ver Figura 2) se asocian, a los

distintos objetos, las funciones/acciones (partes del problema) que estos objetos deberán realizar.

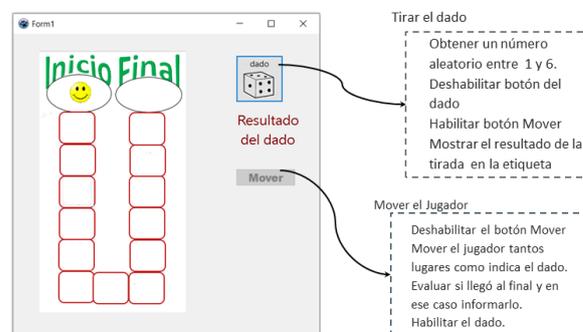


Figura 2. Ejemplo de formulario y diseño de programa (plan)

Esta etapa incluye:

Cada sub-tarea se expresa como una acción con un verbo en infinitivo.

Se describe qué se debe hacer para implementar la tarea. Aquí se puede anotar todo lo que será de utilidad para resolver la sub-tarea (incluso parte del código, tipos de datos, referencias a ejercicios, entre otros).

**Escribir el programa.** Con base en el plan, se escribe en el procedimiento asociado a cada objeto las partes de la solución en formato de comentarios en el código fuente.

Se programa en conjunto una de las funcionalidades. El resto de la programación queda a cargo de las/los estudiantes.

Como afirma [19] “Para escribir un programa, el único medio posible es utilizar un lenguaje de programación. Entonces, el programa constituye el vehículo ideal de explicitación y comunicación de las ideas con las que se concibió la solución expresada”.

**Mirar hacia atrás.** Se revisa la funcionalidad programada en busca de una mejora.

## Resultados y discusión

La propuesta de enseñanza descrita en el punto anterior se implementó por primera vez en 2020 (en el contexto de la Pandemia por la

COVID-19). La asignatura se venía desarrollando de forma presencial y cuenta con un aula virtual en el entorno Moodle como soporte de la presencialidad.

Se revisaron 10 entregas de los planes para la resolución de un ejercicio de la Guía Práctica 3 (promediando la mitad de la cursada y previo al primer parcial). El problema trabaja con dos botones, una etiqueta y una imagen (ver Figura 2 y 3).

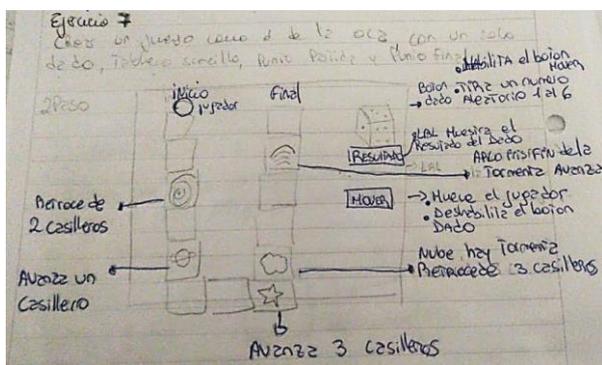


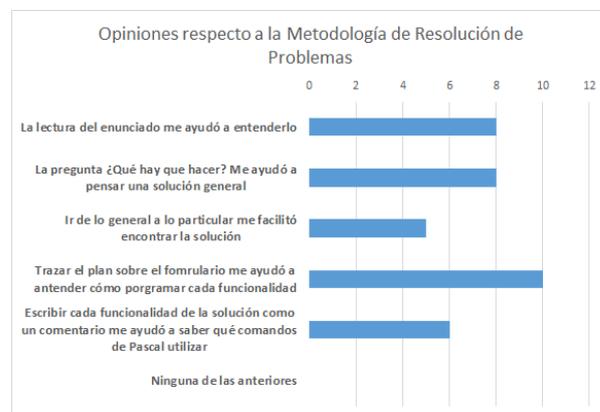
Figura 3. Ejemplo de la propuesta de diseño de un estudiante

Del análisis preliminar de los planes se pudo observar que:

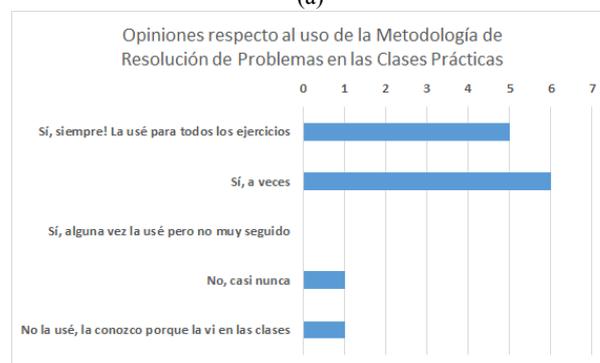
- Las/los estudiantes pudieron identificar y registrar en el diseño parcialmente las funcionalidades. En todos los casos fueron asociadas al objeto correcto (dos botones y una etiqueta).
- Varios estudiantes (4/10) indican en el diseño con qué objetos van a trabajar.
- Un estudiante presenta nodos desconectados en el organizador gráfico, lo que podría indicar que no se está comprendiendo parte del problema.

Por otra parte, en la encuesta final de la cursada se le realizaron dos preguntas sobre la metodología utilizada en la práctica y se obtuvieron 13 respuestas. Como se puede observar en la Figura 4a el trabajo sobre el formulario para diseñar la solución le fue de utilidad a las/los estudiantes, las demás instancias planteadas presentan un bajo nivel de aceptación. En la Figura 4b puede verse que

la mayoría de las/los que respondieron la encuesta se apropiaron de la metodología y la utilizaron en otros problemas de la GP.



(a)



(b)

Figura 4. Resultado de la encuesta a estudiantes sobre la metodología

Dice Lerner (citado en [14]) “Enseñar es promover la discusión sobre los problemas planteados, es brindar la oportunidad de coordinar diferentes puntos de vista, es orientar hacia la resolución cooperativa de las situaciones problemáticas”. La resolución de algunos de los problemas planteados en las Guías Prácticas tiene la pretensión, inicialmente, de mostrar cómo el docente (razonamiento experto) hace uso de la estrategia de resolución de un problema. La propuesta evoluciona hacia una resolución de forma colaborativa donde son las/los estudiantes los que ponen en juego la estrategia de resolución.

En el primer paso, “comprender el problema”, aparece la pregunta ¿Qué hay que hacer? (pregunta de orden cognitivo superior). A

partir de la misma, las/los estudiantes deben interpretar el enunciado y ofrecer una respuesta que pueda resumir en una oración el problema a resolver. Las subsecuentes preguntas se enfocan en “cómo hacerlo”. Las/los estudiantes deben dar cuenta de cómo resolverían el problema, identificar cuáles serían los pasos a seguir (teniendo en cuenta la solución de problemas similares). En este sentido, para [24], las preguntas que contribuyen a la metacognición deberían estar en torno a tres ejes: qué hay que hacer, qué instrumentos necesito para abordar este problema y, cuál es una estrategia adecuada para resolver la tarea.

Con base en lo discutido en el primer paso se diseña el programa. Aquí se hace uso de un organizador gráfico. Se convirtió el formulario en un nodo central desde donde se derivarían todas las funcionalidades que, potencialmente, resolverían el problema. Cada funcionalidad debe iniciar con un verbo en infinitivo (acción) en busca de reflejar la estrategia de solución y lograr una mejor legibilidad.

Para llevar adelante el plan (programación), las/los estudiantes deben identificar qué herramientas del lenguaje les permitirán programar la solución propuesta. Es decir, cómo le “dicen” a la computadora (en su lenguaje) que debe seguir los pasos definidos (metacognición). Si bien no fue posible asociar cada parte del problema con un procedimiento (como plantea Sadosky), dado que no es un concepto que se trabaje en el marco de la asignatura, se optó por dividir el problema en las distintas funcionalidades que se debían implementar. Dice [19] “La herramienta del lenguaje básica para expresar subtareas y estrategias es el procedimiento, y consecuentemente, es una de las primeras herramientas que se presentan y trabajan en los ejercicios.” (p. 10). Así, la división se definió a partir de los objetos sobre los que se debían desarrollar las distintas funcionalidades, cuya programación se hace (en Lazarus) a través de

procedimientos escritos en lenguaje Pascal. Tanto los nombres de los objetos como los procedimientos asociados a los mismos deben tener nombres significativos. “La legibilidad se ve expresada en la actividad de programación a través de la elección de nombres adecuados para las entidades que se escriban en el programa [...]. Se suelen favorecer nombres cortos pero descriptivos, que puedan ser leídos en voz alta en lenguaje natural (castellano, en nuestro caso)” [19, p. 12]. Luego, ya dentro de los procedimientos, la división en subtareas continúa a partir del uso de comentarios en el código fuente. Cada una de las subtareas expresadas en el diseño se colocan como comentarios del código fuente. Hecho esto, se abre un espacio entre los comentarios y se comienza a programar (interactuando con lo registrado en el diseño). Esto implica la traducción de lo escrito en lenguaje natural (comentarios) al lenguaje de programación, realizando una adaptación en función de las herramientas del lenguaje que conocen.

Finalmente “mirar hacia atrás”, en esta etapa se hacen las distintas pruebas y las mejoras o modificaciones que surjan de éstas. En algunos casos, para problemas que pudieran resultar algo complejos, se implementa una versión simplificada del problema y, al mirar hacia atrás, se retoma en enunciado completo y se trabaja sobre la solución definitiva.

## **Conclusiones y trabajos futuros**

En este trabajo se aborda una propuesta didáctica para la enseñanza de la programación enfocada en la resolución de problemas. Esta toma como base la metodología de la Fundación Sadosky y la estrategia de resolución de problemas propuesta por Simon Thompson. La propuesta fue diseñada en 2019-2020 e implementada durante 2020 y se está utilizando, nuevamente, en 2021.

Si bien los resultados son preliminares, y requieren de una nueva implementación, se pudo observar que algunas/os estudiantes se apropian de la metodología y que la propuesta tiene el potencial para actuar como soporte en el proceso de resolución de problemas.

Como trabajos futuros, se deben rediseñar las Guías Prácticas para que los enunciados acompañen apropiadamente a la propuesta (esto se está haciendo para el año en curso). También llevar el marco conceptual al cierre de la guía o al finalizar cada grupo de ejercicios que aborda un concepto, para que se ajusten a lo propuesto en la didáctica por indagación. Lo propio se debería hacer con algunos recursos audiovisuales presentes en el aula virtual, estos deberían estar disponibles al finalizar cada GP.

También, se está evaluando la posibilidad de incluir versiones ejecutables de algunos ejercicios de las GP. Esto con el objetivo de que a través de la ejecución del programa, las/los estudiantes logren identificar los diferentes escenarios o situaciones que el programa debe resolver y las tengan en cuenta en el proceso de resolución.

Por otra parte, se espera contar con más entregas de diseño de planes para tener una mayor cantidad de material para el análisis, así como cortes en diferentes momentos de la cursada, para observar la evolución en la apropiación de la propuesta.

## Bibliografía

- [1] S. Popat y L. Starkey, «Learning to code or coding to learn? A systematic review», *Comput. Educ.*, vol. 128, pp. 365-376, 2019, doi: 10.1016/j.compedu.2018.10.005.
- [2] Ley N. 26.206, *Ley de Educación Nacional*. 2006.
- [3] Presidencia de la Nación, *Decreto 459/2010*. 2010.
- [4] Consejo Federal de Educación, *Resolución CFE N. 263*. 2015.
- [5] Ministerio de Educación y Deportes, *Resolución N. 1536-E*. 2017.
- [6] Consejo Federal de Educación, *Resolución CFE N. 343/18*. 2018.
- [7] G. N. Dapozo, R. H. Petris, C. L. Greiner, M. C. Espíndola, A. M. Company, y M. López, «Capacitación en programación para incorporar el pensamiento computacional en las escuelas», *TE & ET*, vol. no. 18, 2016, Accedido: jul. 30, 2019. [En línea]. Disponible en: <http://hdl.handle.net/10915/58516>.
- [8] Fundación Sadosky, «¿De qué se trata? | Program.AR», *Program.AR*, 2020. <http://program.ar/de-que-se-trata/> (accedido sep. 22, 2020).
- [9] G. J. Astudillo, S. G. Bast, D. Segovia, y L. Castro, «Revisión de propuestas para la enseñanza de la programación», presentado en XII Congreso de Tecnología en Educación y Educación en Tecnología, San Luis, Argentina, 2019.
- [10] G. J. Astudillo y S. Bast, «Enseñanza y aprendizaje de programación. Hacia un estado del arte», *Virtualidad Educ. Cienc.*, vol. 11, n.º 20, Art. n.º 20, 2020.
- [11] G. M. Rodríguez Carrillo, «Enseñanza de la programación de computadoras para principiantes: un contexto histórico», *INVENTUM*, vol. 9, n.º 17, pp. 51-61, 2014, doi: 10.26620/uniminuto.inventum.9.17.2014.51-61.
- [12] M. C. Martínez y M. E. Echeveste, «Representaciones de estudiantes de primaria y secundaria sobre las Ciencias de la Computación y su oficio.», *Rev. Educ. Distancia*, n.º 46, 2015, [En línea]. Disponible en: <https://revistas.um.es/red/article/view/240241>.
- [13] G. J. Astudillo, S. G. Bast, y P. A. Willging, «Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación», *Virtualidad Educ. Cienc.*, vol. 7, n.º 12, pp. 125-142, 2016.
- [14] L. Basabe y E. Cols, «La enseñanza», en

- El saber didáctico*, 1ra ed., Buenos Aires, Argentina: Editorial Paidós, 2007.
- [15] R. Anijovich, S. Mora, y E. Luchetti, *Estrategias de enseñanza: otra mirada al quehacer en el aula*, vol. 1. Buenos Aires, Argentina: Aique, 2009.
- [16] D. Levis, «Enseñar y aprender con informática/ enseñar y aprender informática», en *Medios informáticos en la educación a principios del siglo XXI*, 1ra ed., D. Levis y R. Cabello, Eds. Buenos Aires, Argentina: Prometeo Libros Editorial, 2007, pp. 21-49.
- [17] M. del P. Pérez Echeverría y J. I. Pozo, «Aprender a resolver problemas y resolver problemas para aprender», en *La solución de problemas*, Madrid: Santillana, 1994, pp. 14-50.
- [18] J. I. Pozo, M. d Pérez, J. Domínguez, M. Gómez, y Y. Postigo, *La solución de problemas*. Madrid, España: Santillana Madrid, 1994.
- [19] P. E. Martínez López, «Sugerencias para el dictado del curso La programación y su didáctica. Método Program. AR». Fundación Sadosky, 2016, [En línea]. Disponible en: [http://program.ar/wp-content/uploads/2015/04/GuiaParaCursoProgram.AR\\_1.pdf](http://program.ar/wp-content/uploads/2015/04/GuiaParaCursoProgram.AR_1.pdf).
- [20] G. Polya, *How to solve it*, 2°. New Jersey: Princeton University, 1973.
- [21] S. Thompson, «Where do I begin? A problem solving approach in teaching functional programming», en *Programming Languages: Implementations, Logics, and Programs*, Berlin, Heidelberg, 1997, pp. 323-334, doi: 10.1007/BFb0033853.
- [22] D. J. Barnes, S. Fincher, y S. Thompson, «Introductory Problem Solving in Computer Science», en *5th Annual Conference on the Teaching of Computing*, Centre for Teaching Computing, Dublin City University, Dublin 9, Ireland, ago. 1997, pp. 36-39, Accedido: sep. 30, 2020. [En línea]. Disponible en: <https://kar.kent.ac.uk/21468/>.
- [23] F. Reyes-Cárdenas y K. Padilla, «La indagación y la enseñanza de las ciencias», *Educ. Quím.*, vol. 23, n.º 4, pp. 415-421, 2012.
- [24] P. Meirieu, *La opción de educar: Ética y pedagogía*, 1ra ed. Barcelona, España: OCTAEDRO, 2001.