

## Aprender a programar (¿y a pensar?) jugando

Willging, P.<sup>1,2</sup>, Astudillo, G.<sup>1</sup>, Bast, S.<sup>1</sup>

<sup>1</sup>Facultad de Ciencias Exactas y Naturales – UNLPam, <sup>2</sup>CONICET

[pedro@exactas.unlpam.edu.ar](mailto:pedro@exactas.unlpam.edu.ar), [astudillo@exactas.unlpam.edu.ar](mailto:astudillo@exactas.unlpam.edu.ar), [silviabast@exactas.unlpam.edu.ar](mailto:silviabast@exactas.unlpam.edu.ar)

### Resumen

El éxito o el fracaso en la primera asignatura que cursan los alumnos en sus carreras universitarias suele decidir de manera contundente su permanencia en la institución y su destino profesional. En el área de Informática, los alumnos que se encuentran por primera vez con un lenguaje de programación típicamente tienen el desafío de abordar entre otros el tema de la programación estructurada y la metodología de resolución de problemas. Tanto la temática como la metodología de trabajo son aspectos que los alumnos ingresantes no han trabajado en general lo suficiente en su paso por el nivel medio. Esto conduce a niveles de deserción y fracaso académico muy elevados. Con este contexto, en la Facultad de Ciencias Exactas y Naturales de la UNLPam, se ha generado una propuesta de actividades para alumnos ingresantes que cursarán la asignatura Introducción a la Computación. Esta propuesta, con un enfoque lúdico y de experimentación, incluye un conjunto de actividades basadas en la metodología de resolución de problemas de Polya y el software de animación Scratch. En este trabajo, se presentan los primeros resultados de la experiencia.

**Palabras claves:** resolución de problemas, Scratch, software de animación, ingresantes

### Introducción

Durante las dos semanas previas al inicio del ciclo lectivo, en la Facultad de Ciencias Exactas y Naturales (UNLPam) se pone a disposición de los ingresantes una serie de cursos de apoyo como parte de las jornadas de ambientación universitaria.

Desde la cátedra de Introducción a la Computación se propuso un “Curso de Resolución de Problemas” con los siguientes objetivos:

- Identificar los datos e incógnitas de un problema dentro de un enunciado.
- Utilizar estrategias adecuadas para la resolución de un problema.
- Incorporar la noción de algoritmo.
- Identificar primitivas de entrada, salida, selección, y repetición y utilizarlas apropiadamente para el diseño del algoritmo que resuelva un problema

Las actividades propuestas para los alumnos, que se desarrollan durante los primeros meses del año, incluyen: diagnóstico, ayudas ajustadas, evaluación formativa, e integración de tutores académicos. Para el diseño de los problemas presentados a los alumnos (implementados con la metodología de resolución de problemas como guía) se utilizó por primera vez el software de animación Scratch, por su potencial pedagógico y motivador de la creatividad. Esta es una experiencia innovadora que se aplica con el propósito de lograr una articulación más efectiva entre los niveles secundarios y universitarios, motivar a los ingresantes y propiciar en ellos el desarrollo de nuevas estrategias de aprendizaje. En las siguientes secciones, se dará el marco teórico que sustenta la propuesta, se describirán los métodos y procedimientos empleados, y se discutirán los resultados obtenidos.

### Marco Teórico

#### Evaluación formativa

La evaluación que se realiza durante el proceso de aprendizaje o evaluación formativa es un concepto introducido por Scriven (1967)

y su función es obtener información acerca del estado de aprendizaje de cada estudiante y, a partir de ello, tomar decisiones que mejoren el desarrollo de dicho proceso. La información que arroja este tipo de evaluación, es de utilidad obviamente para el docente, pero también debe ponerse a disposición de los estudiantes, de forma que también puedan hacerse cargo de su propio proceso de aprendizaje.

Responde a características tales como: se aplica durante el desarrollo de una unidad de aprendizaje, no necesariamente debe contar con una calificación, requiere de momentos de intercambio, en los cuales los estudiantes puedan recibir explicaciones acerca de sus problemas y errores, puede presentarse en distintos formatos tales como informes, dramatizaciones, conversaciones con los estudiantes y no necesariamente debe tener la forma de un cuestionario.

En este tipo de evaluación los errores de los alumnos se estudian especialmente, ya que son un insumo muy importante que posibilita determinar las dificultades que presentan en la realización de las tareas propuestas y de esta manera, poner a su disposición los medios para poder superar estos obstáculos. También se resaltan las tareas exitosas para motivar a los estudiantes.

La evaluación diagnóstica, considerada como una forma de evaluación formativa, tiene como objetivo obtener información del grado de conocimientos previos, así como también de las habilidades y capacidades cognitivas en que se encuentran los alumnos. Esta evaluación se realiza antes de comenzar con el proceso de enseñanza y aprendizaje y la información que arroje será el punto de partida para comenzar a trabajar en el logro de los objetivos planteados, detectar carencias que puedan dificultar el proceso, diseñar actividades que permitan nivelar los aprendizajes.

Sus características son: no califica al estudiante, no necesariamente debe ser una prueba, puede llevarse a cabo de forma individual o grupal y debe existir una devolución a los estudiantes.

### **Las ayudas ajustadas**

El concepto de ayuda ajustada, proviene del constructivismo e implica un conjunto de acciones de planificación, ejecución y evaluación permanente que el docente debe efectuar antes, durante y después de la clase, cuyo objetivo es promover la comprensión y la autonomía del estudiante en la realización de la tarea planteada. Pero estas intervenciones, o ayudas, entre las que pueden mencionarse, entre otras: explorar conocimientos previos, dar pistas, corregir errores, organizar y estructurar la clase, deben adaptarse a los destinatarios. Es claro que una determinada ayuda puede resultar apropiada para una situación de aprendizaje particular y con un grupo-clase específico, y fracasar en otro caso. Es por esto que hablamos de ayudas ajustadas. Según afirma Beas (2003) debe tenerse en cuenta que:

- La enseñanza no es lineal: una misma forma de intervención puede servir para unos alumnos y en un momento, y no servir para otros.
- Las ayudas deben ser muy variadas: sobre una misma situación de enseñanza se deben disponer ayudas de distinta naturaleza para que se ajusten a las diversas características de los individuos o grupos.
- Las ayudas dependen de la fase de aprendizaje en que se encuentre el alumno: no es lo mismo si está al inicio, en el proceso o al final del aprendizaje.

La idea de ajuste esta fuertemente ligada a la adaptación de la ayuda al contexto particular. Como afirma (Onrubia, 2007, pág. 102):

“La consideración de la enseñanza como ayuda al proceso de aprendizaje tiene [...] una consecuencia [...] la delimitación del ajuste de dicha ayuda al proceso constructivo que realiza el alumno.”

Las ayudas ajustadas deben plantear desafíos para el alumno. Esto es, se debe apuntar a aquello que el alumno no conoce o no domina

completamente, debe implicar un esfuerzo intelectual para el alumno. Pero, además, debe partir de los conocimientos previos del alumno. Por tanto, “la ayuda ajustada supone retos abordables para el alumno, [... una] combinación de sus propias posibilidades y de los apoyos e instrumentos que reciba del profesor” (Onrubia, 2007, pág.103).

Cabe aclarar que la propuesta de las ayudas ajustadas se enmarca dentro de lo que Vygotski (1979) denominaba Zona de Desarrollo Próximo (ZDP) que puede definirse como “la distancia entre el nivel de resolución de una tarea que una persona puede alcanzar actuando independientemente y el nivel que puede alcanzar con la ayuda de un compañero más competente o experto en esa tarea”. En este sentido afirma Onrubia (2007), “ofrecer una ayuda ajustada al aprendizaje escolar supone crear ZDP y ofrecer asistencia y apoyos en ellas”.

### **La estrategia de resolución de problemas**

La resolución de problemas implica, generalmente, un esfuerzo intelectual y el diseño de un conjunto de pasos que permitan llegar a la solución del problema. El matemático George Polya (1973) planteó en su libro “How to solve it” una estrategia para la resolución de problemas que está dividida en cuatro pasos: entender el problema, diseñar un plan, ejecutar el plan, y mirar hacia atrás.

Como afirma Polya, lo peor que puede suceder es que el alumno se embarque en la solución del problema sin entenderlo. Por tanto, la primera fase es **entender el problema**. Lo que se debe hacer en esta fase es comprender el enunciado. Leerlo varias veces y buscar las palabras que no se conocen. Esto es, garantizar la comprensión total y absoluta del enunciado. Luego se deben identificar las principales partes del problema: lo que no se conoce, los datos y la condición. En esta etapa sería importante que, de ser posible, se realicen dibujos, bosquejos y/o gráficas que ayuden a visualizar el problema y aporten a su comprensión. También, suele ser útil darle nombre a los objetos de manera de poder

manejarlos a través de alguna simbología (por ejemplo, para plantear ecuaciones).

En esta etapa la función del docente es diseñar estrategias que ayuden al alumno a identificar y extraer la información del problema.

Polya sugiere que se pueden realizar preguntas como: ¿Qué es lo que no se conoce en el problema? ¿Cuáles son los datos? ¿Cuál es la condición?

La segunda fase, **diseñar un plan**, es la más complicada. Requiere de un gran esfuerzo intelectual y debe convertirse en una guía que lleve a resolver el problema. Es importante tener paciencia ya que las ideas que van a llevar a la construcción del plan pueden ir surgiendo gradualmente, y los primeros intentos pueden resultar algo frustrantes. En esta etapa es apropiado tratar de identificar problemas similares y analizar si la solución (o parte de ella) se puede adaptar al problema en cuestión. Además, si no es posible resolver el problema en su totalidad, se puede intentar resolver un problema similar de menor complejidad, o bien, dividir el problema en problemas más pequeños (de menor complejidad) e ir resolviendo por partes. El papel del docente en esta fase es provocar en el alumno el surgimiento del plan a través de sugerencias y preguntas.

Las preguntas que sugiere Polya para esta etapa son: ¿Conoces algún problema relacionado? Hay un problema relacionado y resuelto antes; ¿Podrías utilizarlo? ¿Podrías reformular el problema? ¿Utilizaste todos los datos? ¿Has utilizado toda la condición?

La tercera fase, **ejecutar el plan**, es más sencilla que la anterior, solo hay que tener la precaución de seguir meticulosamente el plan trazado. El plan ofrece una guía que conduce a una posible solución. El docente debe tener muy poca presencia en esta etapa. Debe limitarse a procurar que el alumno siga el plan propuesto, y que sea su propio plan.

La última etapa es la denominada **mirar hacia atrás**, y es la que habitualmente los alumnos dejan de lado. Una vez que el problema está resuelto, ellos suelen pasar al próximo sin darse la oportunidad de revisar la solución. Para Polya (1973, pág.14) “es la fase más

importante e instructiva del proceso”. Esta fase tiene como objetivos no sólo la búsqueda de posibles errores, sino también aprender del proceso. El docente debe asegurarse de que el proceso de revisión se realice. Esto puede hacerse pidiendo al alumno que imagine otros contextos donde el plan puede volver a utilizarse exitosamente.

Algunas preguntas para esta etapa podrían ser: ¿Se pueden chequear los resultados? ¿Se puede llegar a un resultado distinto? La solución: ¿Puede verse a primera vista? ¿Podría utilizar los resultados o el método, para algún otro problema?

La metodología propuesta por Polya, claro está, no garantiza alcanzar la solución del problema pero ofrece una forma de organizar la resolución del problema. La forma precisa en el que el problema será resuelto depende, en gran medida, de las habilidades cognitivas de quien lo resuelve y de la práctica que tiene en la resolución del problemas.

### **El software de animación Scratch**

Scratch es un lenguaje de programación con el que se pueden crear historias interactivas, animaciones, juegos, música y arte; los cuales pueden compartirse alojándolos en el sitio web <http://scratch.mit.edu>. Los usuarios de Scratch crean y comparten proyectos al mismo tiempo que aprenden importantes ideas matemáticas y computacionales; y más importante aún, aprenden a pensar creativamente, a razonar sistemáticamente y a trabajar colaborativamente (Resnik et al., 2009). Scratch es un proyecto desarrollado por el Lifelong Kindergarten Group en el Laboratorio de Medios de MIT, con apoyo financiero de la National Science Foundation y algunas empresas comerciales importantes. Ha sido desarrollado con Squeak como lenguaje de implementación, las versiones disponibles funcionan en Windows, Mac, y también en Linux por medio de la utilización de una máquina virtual. El software viene con una librería de medios extensa y con una colección de “Proyectos de Ejemplo”, los cuales los aprendices pueden utilizar para experimentar e iniciar los primeros proyectos propios. La

mayoría de las computadoras tienen memoria suficiente para correr Scratch (en los equipos más viejos puede correr más lentamente). Se necesitan tan solo 120 megabytes de espacio libre en el disco para instalarlo.

El grupo creador de Scratch toma como antecedentes de su desarrollo las experiencias del lenguaje Logo, que fuera presentado por Seymour Papert en su libro *Mindstorms* (1980), y otros proyectos relacionados a las animaciones y software de programación como Alice (Kelleher & Pausch, 2007). La implementación de Scratch fue planificada de modo que tal que permitiera a cualquiera, niño, adolescente o adulto sin experiencia en programación, aprender a crear sus propias animaciones. Para lograrlo, sus creadores establecieron tres principios fundamentales: hacerlo más flexible, significativo y con mayor interacción social que otros entornos de programación existentes.

La gramática de Scratch se basa en un conjunto de “bloques gráficos de programación” que se ensamblan para crear programas, imitando lo que hacen los niños cuando juegan con los bloques de LEGO (Resnik, 2007). Se puede experimentar con los bloques apilándolos de diferente forma y en diferentes secuencias, y de modo instantáneo ver cual es el efecto de los cambios y modificaciones. En palabras de Resnik: “Los bloques de Scratch se diseñan para que solamente encajen de maneras que hagan sentido sintácticamente. Las estructuras de Control tales como, por siempre y repetir, tienen forma de C y con ella se quiere sugerir que los bloques deben ponerse dentro de estas. Los bloques con valores de salida tienen formas acordes con el tipo de valor que retornan: óvalos para valores numéricos y hexágonos para valores Booleanos. Los bloques condicionales tales como si y repetir hasta que, tienen espacios de forma hexagonal, para indicar que se requiere poner en ellos un Booleano.”

La interfaz de Scratch (Figura 1) imita un escritorio físico donde se pueden dejar bloques extras que se pueden necesitar para otro experimento. En este ambiente es aceptable ser

un poco desordenado y experimentar. Con Scratch es posible experimentar de modo desordenado, pero también crear de modo planificado y ordenado. Existen ya varias experiencias de utilización de Scratch como recurso pedagógico (Crook, 2009; Peppler & Kafai, 2007), y no es difícil anticipar que será utilizado en muchas más experiencias de docencia e investigación.



**Figura 1. Ambiente de trabajo de Scratch**

Un aspecto importante en el proyecto Scratch, es el concepto de compartir las producciones en la web. Esto se estimula desde la misma interfaz del software, que tiene un botón específico para subir los proyectos a la web. Los proyectos pueden visualizarse en cualquier computadora por medio del plugin de Java. En este momento, el sitio web de Scratch cuenta con casi un millón de proyectos, amparados por la licencia Creative Commons, y tiene casi medio millón de usuarios distribuidos por todo el planeta. Esto demuestra a las claras el éxito de la propuesta. No obstante, sus creadores tienen mayores expectativas aún: “Es necesario expandir la noción de “Fluidez o competencia digital” para que incluya diseñar y crear, no solamente navegar e interactuar. Solo en ese momento tendrán iniciativas como Scratch la posibilidad de desplegar todo su potencial” (Resnik et al., 2009).

## Metodología

El curso/taller de resolución de problemas consiste en un conjunto de actividades propuestas a los alumnos que se describirán

agrupándolas en cuatro categorías: diagnóstico, ayudas ajustadas, evaluación formativa e integración de tutores académicos.

### Diagnóstico

Esta actividad permite identificar los conocimientos previos de los alumnos acerca de:

- ¿Qué es un problema?
- ¿Cuáles son los pasos a seguir para resolver un problema?
- ¿Cómo puede ayudar la computadora a resolver un problema?
- ¿Qué es un lenguaje de programación?

Para realizar la actividad diagnóstica se utiliza la herramienta *Retroalimentación (Feedback)* de la plataforma Moodle<sup>1</sup>. Esta actividad al estar disponible en Internet puede ser completada por los ingresantes inmediatamente después de su inscripción a la carrera.

### Ayudas ajustadas

Aquí se incluyen un grupo de actividades que les permitan a los alumnos construir las nociones de: problema, estrategias para la resolución de problemas, y heurísticas. Además, identificar los mecanismos que les permitan traducir el plan a un lenguaje que pueda ser procesado por una computadora. Las actividades planificadas para esta etapa son:

- Reconocer las estrategias que utiliza un experto para resolver un problema.
- Reconocer estrategias para resolver un problema a través de la revisión de ejemplos de problemas resueltos.
- Resolver ejercicios simples. Estos son similares a los propuestos en los apartados anteriores, donde los alumnos pueden aplicar las estrategias desplegadas (implícitamente) en las resoluciones.
- Leer el material bibliográfico de apoyo basado en la metodología de resolución de problemas de Polya.
- Resolver problemas a través del diseño de algoritmos, utilizando Scratch.

<sup>1</sup> Moodle es una plataforma para desarrollo de cursos virtuales (ver moodle.org)

Se provee un breve tutorial (presencial) sobre el manejo del software. Para luego continuar con un conjunto de problemas, de complejidad creciente, que se pueden resolver utilizando la aplicación.

### Evaluación formativa

Para cada actividad propuesta a los alumnos se les realiza una devolución de manera que puedan reconocer sus errores y, de ser necesario, se les propone que revisen y rehagan las soluciones.

### Integración de tutores académicos

Las actividades cuentan con la colaboración de tutores académicos<sup>2</sup> de la materia. Al colaborar en la resolución de las actividades junto a los ingresantes y ser considerados por ellos un par más experto se propicia un clima de confianza y horizontalidad.

## Implementación y Procedimientos

Para la implementación del Curso de Resolución de Problemas se creó un curso virtual sobre la plataforma Moodle. El curso consta de cinco bloques temáticos: (1) ¿Será posible resolver este acertijo?, (2) Los Problemas, (3) Resolución de problemas con Scratch I, (4) Resolución de problemas con Scratch II, y (5) Evaluación. Los bloques fueron presentados (descubiertos) de manera secuencial, utilizando la herramienta de la plataforma virtual que permite hacer visible o invisible un bloque de acuerdo a lo que el instructor decida. Los recursos de la plataforma utilizados para el diseño fueron: Cuestionarios, Foros, Glosarios, Lecciones, Tareas y Retroalimentación (módulo para encuesta). Además, se utilizaron videos, imágenes, documentos PDF y páginas web.

El bloque temático 1 (Figura 2) contiene dentro de él dos enlaces a videos de YouTube que muestran fragmentos de la película

española *La habitación de Fermat*<sup>3</sup> en los que se presentan a los personajes dos acertijos (uno en cada enlace). Además, para cada video hay un foro donde los alumnos deben hacer una propuesta de solución. El tipo de foro elegido es *preguntas y respuestas* que sólo muestra las respuestas si previamente se realiza un aporte al foro. Esto permitió que los alumnos hicieran su primer aporte sin contaminarse con las respuestas de los demás. Para luego, tener acceso a las respuestas y contrastar su aporte con el de los demás alumnos. Cada aporte tuvo una respuesta por parte de los docentes, con el fin de alentar a quienes daban una respuesta correcta y de guiar a un mejor planteo o resolución a quienes no respondían correctamente. Este bloque se extendió durante 2 semanas, poniéndose visible un acertijo por semana.



**Figura 2.** Parte del Bloque temático 1 del curso.

El segundo bloque, **Los Problemas** (Figura 3), se hizo visible a partir de la tercera semana de iniciado el curso. Cuenta con un documento PDF, diseñado por los integrantes de la cátedra, donde se explica el método de resolución de problemas de Polya. Los

<sup>2</sup> Los tutores académicos son alumnos de la carrera de Informática que ya han cursado la asignatura en años previos.

<sup>3</sup> Piedrahita L. & Sopeña R. (2007). *La habitación de Fermat*. Thriller. Notro Films y BocaBoca.

alumnos debían leer el documento para interiorizarse del método y tenían un foro para realizar consultas. Como ejemplo de aplicación del método de Polya, los alumnos pueden seguir dos lecciones (resolución por expertos). Estas lecciones cuentan además con preguntas que el alumno debe responder para acceder al paso siguiente. En consonancia con lo que Polya plantea para la primera fase de la resolución de un problema (comprender claramente el enunciado), los alumnos cuentan con un glosario y enlaces a la página de la Real Academia Española y al diccionario de sinónimos Wordreference.com para resolver sus dudas si es que tienen dificultades en la comprensión de los enunciados de los problemas presentados.



**Figura 3. Bloque 2 del curso.**

La apertura del tercer bloque (Figura 4), se realizó la cuarta semana del curso y coincide con el primer encuentro presencial con los alumnos. Este bloque cuenta con un enlace a una galería de Scratch con una propuesta de solución para uno de los problemas planteados durante la primera semana. También, tiene asignada una carpeta para recolección de archivos, para permitirles a los alumnos subir sus propias producciones. Como recursos adicionales se disponen guías para Scratch en formato PDF, enlaces al sitio oficial de Scratch, códigos fuente de Scratch y algunas imágenes que los alumnos pueden incluir en la resolución de los problemas.

En el encuentro presencial se les muestra a los alumnos el ejemplo de la galería donde aparecen, implícitamente, las estrategias utilizadas por los expertos en la resolución del

problema. Luego, se da una muy breve descripción del ambiente de trabajo de Scratch, y se analizan un par de programas sencillos diseñados en Scratch. Uno de los programas está terminado y es analizado en conjunto con el grupo-clase y luego probado por los alumnos, mostrando nuevamente como los expertos resuelven un problema. El otro ejemplo, sólo tiene una parte funcional y los alumnos deben analizar el funcionamiento y completar la parte que falta (adaptar una solución conocida a otro problema). Como tarea se les deja a los alumnos la solución de uno de los problemas planteados en las lecciones en el bloque 2. Esto tiene como finalidad que los alumnos se habitúen al uso de la herramienta y no sumar complejidad a través de nuevos problemas. Los programas entregados son evaluados (no calificados) y se les hace una devolución personalizada (a través de la plataforma), y se invita a aquellos alumnos cuyo programa no está del todo correcto a volver a remitirlo. Conformando así otra instancia de evaluación formativa.



**Figura 4. Parte del Bloque 3 del curso.**

El quinto bloque, se lleva adelante en coincidencia con el segundo encuentro presencial. En él se ponen a disposición de los alumnos un par de ejemplos incompletos que muestran una resolución de un problema más

complejo utilizando Scratch. Estos ejercicios incorporan variables y estructuras condicionales, que en los anteriores no se abordaban. En la clase presencial se discuten los ejemplos con el grupo-clase y se invita a los alumnos a completarlos. Aquí se trabaja a partir de preguntas para inducir el surgimiento de posibles soluciones a partir del intercambio de ideas entre el grupo-clase.

Como tarea, se tiene un documento de texto con ejercitación. Para hacer consultas los alumnos cuentan con un foro. Dentro de los ejercicios propuestos deben elegir uno para entregar subiendo el resultado por medio de la plataforma Moodle. Aquí, como se viene haciendo en la demás actividades, se hace una devolución personalizada al alumno, invitando a revisar y re-enviar en caso de ser necesario.

El último bloque, presenta una evaluación realizada con un Cuestionario de Moodle, para identificar los logros en el aprendizaje de los alumnos. Además, el bloque contiene una encuesta de opinión respecto del nivel de satisfacción con la experiencia desde el punto de vista del alumno.

## Resultados y Discusión

Por cuestiones de espacio, en esta sección discutiremos solo una parte de los resultados obtenidos hasta el momento.

De los 81<sup>4</sup> alumnos inscriptos para cursar Introducción a la Computación que podían optar por participar del curso 54 participaron de los encuentros presenciales y 63 alumnos se inscribieron en el curso en Moodle. De estos 63 participantes, 46 son mujeres, y 17 varones. Respecto a su procedencia, 41 son locales (de Santa Rosa) y 22 de otras localidades. Los alumnos participaron en esta propuesta de manera totalmente voluntaria.

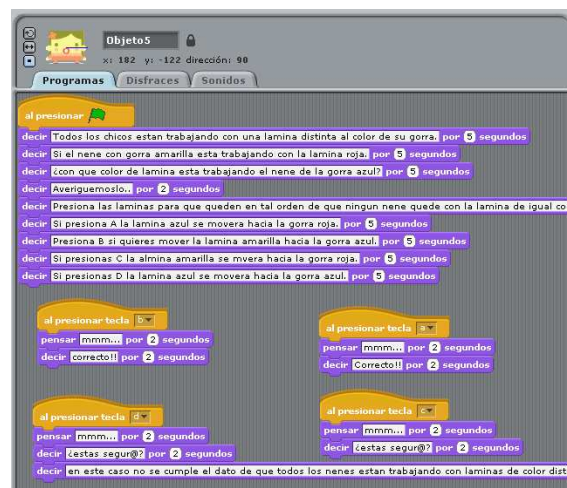
Respecto a la participación en la plataforma virtual, durante las primeras dos semanas fue discreta, contándose 17 aportes, de los 30 alumnos que se habían inscripto en el curso (la cantidad de ingresantes que se incorporaron al curso fue incrementándose después de las dos

primeras semanas). Una vez comenzadas las clases presenciales (ya con los 63 inscriptos en Moodle) no se permitieron más aportes a los foros de la primera semana (bloque de acertijos) debido a que se presentaban las soluciones en clase.

El problema planteado como tarea y que tenían resuelto a través de una Lección (en Moodle) se enunció de la siguiente manera:

*Tres niños están haciendo una investigación sobre plantas y vuelcan la información sobre tres láminas, una de color amarillo, otra roja y otra azul. Los chicos tienen gorras de los mismos colores de las láminas. Después de observar un momento, el niño de gorra amarilla, que trabajaba sobre la lámina roja, le dice a los otros: “¡Miren lo que descubrí! Cada uno de nosotros trabaja sobre una lámina de color distinto a su gorra” ¿De qué color es la lámina en la que trabaja el nene con gorra azul?*

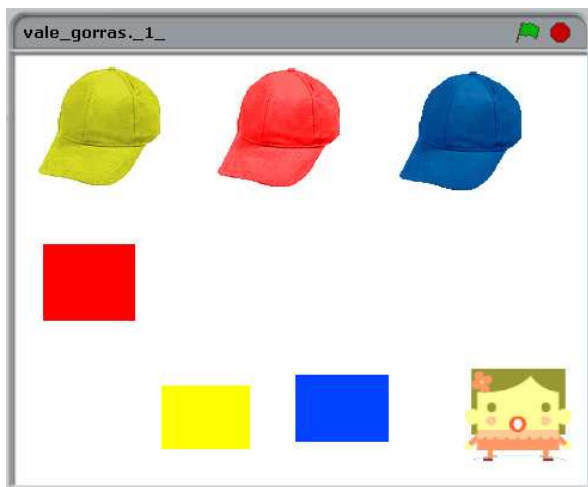
Cuando se les presentó este problema los alumnos solo contaban con aproximadamente una hora de práctica con el programa Scratch y pudieron resolverlo por ellos mismos fuera del horario de la clase presencial. Como ejemplo del tipo de resolución a la que arribaron, se muestra (Figura 5 y Figura 6) una respuesta publicada por un alumno donde propone un algoritmo en el que no sólo representa la solución, sino que además quien opera el programa debe tener la precaución de elegir las opciones correctas.



**Figura 5. Resolución de un alumno del problema de las gorras (código fuente)**

<sup>4</sup> Datos finales según consta en el sistema SIU Guaraní.





**Figura 6. Solución propuesta por un alumno del problema de las gorras (pantalla de ejecución)**

## Conclusiones

Los datos recolectados en esta novel experiencia nos permiten afirmar que los objetivos propuestos han sido logrados con suficiencia. Pudimos comprobar, como efecto positivo no anticipado, que se generó un clima de competencia y juego, con mucha motivación y entusiasmo, tanto de parte de los alumnos como de los docentes participantes en esta intervención.

En lo que respecta a la utilización de la herramienta Scratch, los alumnos lograron utilizarla de modo casi inmediato, necesitando solamente una rápida descripción de la interface. Se dio el caso de un alumno que publicó una solución antes del encuentro presencial. Esto nos confirma el bajo piso del software, alumnos sin experiencia en programación puede diseñar y crear (programar) sus propias animaciones. Por años de experiencia de participar en el dictado de cursos introductorios de programación, sabemos que esto es mucho más difícil de lograr en cualquier otro entorno de programación.

El “problema de las gorras”, cuya sencilla resolución debía ser representada por los alumnos en Scratch, resultó una actividad muy reveladora - pudimos constatar que los

alumnos a partir de una breve inmersión en el software han podido representar la solución al problema utilizando una secuencia ordenada de pasos. Esto nos conduce a pensar que una noción (si bien informal) de algoritmo puede estar siendo incorporada por los alumnos.

Estos resultados tan alentadores nos impulsan a continuar con la propuesta, enriqueciéndola y extendiéndola, y también difundiéndola para que se convierta en una estrategia integrada a los cursos introductorios de programación.

## Bibliografía

**Beas, J., Santa Cruz, J., Thomsen, P., & Utreras, S.** (2003). Enseñar a pensar para aprender mejor (3ª ed.). Santiago: Universidad Católica de Chile.

**Crook, S.** (2009). Embedding Scratch in the classroom. Disponible en línea

<http://scratch.redware.com/scratch-in-the-classroom.html>

**Kelleher, C. & Pausch, R.** (2007). Using storytelling to motivate programming. *Communications of the ACM* 50(7), 58–64.

**Monroy-Hernández, A., & Resnick, M.** (2008). Empowering kids to create and share programmable media. *Interactions*, March-April 2008.

**Onrubia, J.** (2007). Enseñar: crear zonas de desarrollo próximo e intervenir en ellas. En *El constructivismo en el aula* (18º ed.). Barcelona: Editorial Graó.

**Papert, S.** (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York : Basic Books.

**Peppler, K., & Kafai, Y. B.** (2007). From SuperGoo to Scratch: Exploring Media Creative Production in an Informal Learning Environment. *Journal on Learning, Media, and Technology*, 32(2), 149-166.

**Peppler, K. & Kafai, Y. B. (2007).** Collaboration, Computation, and Creativity: Media Arts Practices in Urban Youth Culture. In C. Hmelo-Silver & A. O'Donnell (Eds.), Proceedings of the Conference on Computer Supported Collaborative Learning, New Brunswick, NJ.

**Polya, G. (1973).** How to solve it (2<sup>o</sup> ed.). New Jersey: Princeton University.

**Resnick, M. (2007).** All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. Proceedings of the SIGCHI Conference on Creativity and Cognition, Washington, D.C.

**Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009).** Scratch: Programming for All. Communications of the ACM, 52 (11), 60-67.

**Scriven, M. (1967).** The methodology of evaluation. In R. W. Tyler, R. M. Gagné, & M. Scriven (Eds.), Perspectives of curriculum evaluation, 39-83. Chicago, IL: Rand McNally.

**Vigotsky, L. S. (1979).** El desarrollo de los procesos psicológicos superiores. Barcelona: Crítica.