

Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile

Cristian L. Vidal⁽¹⁾, Carlos Cabezas⁽²⁾, José H. Parra⁽³⁾ y Leopoldo P. López⁽⁴⁾

(1) Departamento de Computación e Informática, Facultad de Ingeniería, Universidad de Playa Ancha, Avenida Leopoldo Carvallo 270, Playa Ancha, Valparaíso-Chile (e-mail: cristian.vidal@upla.cl)

(2) Departamento de Matemática, Facultad de Ciencias Básicas, Universidad Católica del Maule, Avenida San Miguel 3605, Talca-Chile (e-mail: ccabezas@ucm.cl)

(3) Departamento de Formación Escolar Inicial, Facultad de Ciencias de la Educación, Universidad Católica del Maule, Avenida San Miguel 3605, Talca-Chile (e-mail: jparra@ucm.cl)

(4) Instituto de Investigación y Desarrollo Educacional, IIIDE, Universidad de Talca Campus Lircay, Avenida Lircay S/N, Talca-Chile (e-mail: llopez@utalca.cl)

Recibido Oct. 2, 2014; Aceptado Nov. 27, 2014; Versión final recibida Feb. 14, 2015

Resumen

El principal objetivo de este trabajo es presentar los resultados de experimentos de enseñanza y aprendizaje del uso de un lenguaje de programación orientado a objetos. Este experimento fue realizado con estudiantes de un colegio de la ciudad de Viña del Mar y uno de la ciudad de Linares, Chile. El experimento mostró la potencial efectividad en el desarrollo del pensamiento lógico y algorítmico en estudiantes de enseñanza media. Este trabajo describe las principales propiedades del lenguaje de programación Scratch, así como sus ventajas potenciales, respecto a lenguajes de programación tradicionales, para el desarrollo del pensamiento lógico y algorítmico. Además, se muestra que para realizar cada uno de los experimentos es necesario hacer ensayos con la herramienta Scratch y su interfaz. Finalmente, este trabajo presenta los resultados de los experimentos aplicados y resalta el potencial impacto positivo del uso de Scratch y el desarrollo del pensamiento algorítmico de estudiantes en Chile, antes de entrar en la universidad.

Palabras clave: lenguaje Scratch, pensamiento algorítmico, estudiantes secundarios, lenguaje de programación

Practical Experiences for Using the Programming Language Scratch to Develop Algorithmic Thinking of Students in Chile

Abstract

The main objective of this article is to present the results of experiments in teaching and learning using an object-oriented programming language. This experiment was performed on students of two secondary schools, one from the city of Viña del Mar and another one from the city of Linares, both in Chile. The experiment showed the potential effects of developing the logic and algorithmic thinking of high school students. This article describes the main properties of the Scratch programming language, as well as its potential advantages to develop the logic and algorithmic thinking of students in Chile, compared to traditional programming languages. Furthermore, this article shows that to perform each of the experiments, practices with the Scratch and its interface is needed. Finally, this work presents the results of the applied experiments and highlights the potential positive impact of using Scratch and developing the algorithmic thinking of students in Chile, before entering the university.

Keywords: Scratch language, algorithmic thinking, high school students, programing language

INTRODUCCIÓN

Sin lugar a dudas, es relevante para el ser humano, el desarrollo del pensamiento lógico para analizar y solucionar problemas y situaciones de su vida diaria (CVNE, 2014). Particularmente, su uso es esencial para el análisis, comprensión y solución de problemas computacionales, relacionados con la cultura de nuestra época y, por lo mismo, es imperativo su desarrollo en las presentes y futuras generaciones. En este aspecto, el desarrollo del pensamiento lógico y algorítmico es, usualmente, mediado por el uso de lenguajes de programación cuyo aprendizaje requiere un dominio de la sintaxis y semántica de dichos lenguajes (Joyanes Aguilar, 2003), hecho que dificulta el acceso y el uso de los sistemas computacionales.

Desarrollar el razonamiento lógico y algorítmico no solo permite analizar problemas y entregar soluciones en el ámbito computacional, sino en otras situaciones de la vida diaria tan variadas como las artes (Mora, 2007) o la empresa (Pacheco, 2007) respectivamente. Por ejemplo, tener una reunión a una hora determinada para realizar una tarea. Para cumplir con el propósito de estar a la hora en dicha reunión, sin estar atrasado ni muy adelantado, es necesario plantear una solución lógica y algorítmica al problema, la figura 1 presenta un algoritmo usando pseudocódigo de solución a éste.

Por otro lado presentar alternativas didácticas para estimular el desarrollo del pensamiento lógico y algorítmico en niños y estudiantes representa una meta declarada de los sistemas educativos, como se justifica más abajo en la presentación del pensamiento algorítmico. Si bien con el uso de lenguajes de programación tradicionales, esta meta no parece del todo alcanzable, debido a las dificultades señaladas anteriormente, existen lenguajes de programación animados como Scratch y Alice (Scratch, 2014; Alice, 2014), que reducen el costo del aprendizaje de reglas sintácticas y semánticas de los lenguajes de programación tradicionales, y facilitan, a través del uso de elementos multimedia como imágenes y sonido, la visualización de elementos algorítmicos tales como movimiento, condiciones y repetición de acciones. Estos lenguajes facilitan diferentes formas de aprendizaje tales como e-aprendizaje y t-aprendizaje descritas en (Acevedo et al., 2010). Justamente, en la búsqueda de actividades didácticas que estimulen el desarrollo del pensamiento lógico y algorítmico de los estudiantes, se realizó un experimento de enseñanza-aprendizaje y aplicación de Scratch con estudiantes de una escuela de Viña del Mar y una de Linares, Chile, con el cual se busca verificar la viabilidad de estimular con este medio, el desarrollo del pensamiento lógico y algorítmico en niños chilenos.

Este trabajo se organiza en tres secciones: en la primera ANÁLISIS LÓGICO ALGORÍTMICO Y SCRATCH, se conceptualiza el pensamiento algorítmico y se ejemplifican las ideas claves de este tipo de pensamiento. Además se detallan los principales elementos de Scratch que aportan a su desarrollo y utilización en el mundo; la sección 2, ALGORITMOS Y EDUCACIÓN está dedicada a destacar la importancia que atribuyen al pensamiento algorítmico según las bases curriculares en el área de la matemática 2013 del MINEDUC (2014) y CURRÍCULO (2014), Chile, además de describir los fundamentos didácticos para la educación así como el juego como estrategia de enseñanza, con la finalidad de motivar el uso de Scratch en educación; en la sección 3, APLICACIÓN Y RESULTADOS se describe el experimento de enseñanza-aprendizaje y aplicación de Scratch con estudiantes de una escuela de Viña del Mar y estudiantes de un colegio de Linares, Chile, junto con presentar un análisis de los resultados obtenidos; finalmente, una sección de CONCLUSIONES; donde se describen principales ideas obtenidas por este trabajo así como una lista de trabajos actuales y futuros.

```
Algoritmo Llegar_a_la_Hora
  Entrada: Día y Hora de Reunión
           Dirección
  Salida: Establecer_Alarma
Inicio
  Leer Día, Hora, Dirección
  Tiempo = Tiempo_a_Dirección + 30
  Fijar Alarma a Día y (Hora - Tiempo)
Fin
```

Fig. 1: Algoritmo ejemplo para establecer horario de alarma.

ANÁLISIS LÓGICO ALGORÍTMICO Y SCRATCH

Pensamiento Algorítmico

Se denomina algoritmo a un grupo finito de operaciones organizadas de manera lógica y ordenada que permite solucionar un determinado problema (RAE, 2014). Se trata de una serie de instrucciones o reglas establecidas que, por medio de una sucesión de pasos, permiten arribar a un resultado o solución. Por ejemplo, en el área de la matemática destacamos dos algoritmos fundamentales descritos en “Los Elementos” de Euclides, escrito alrededor del año 300 a.C., los cuales son el algoritmo de la división que se utiliza en el cálculo del cociente y el resto al dividir un número entero por otro y, el algoritmo euclidiano usado para el cálculo del máximo divisor común (Coutinho, 2005). El pensamiento algorítmico ayuda a los estudiantes a pasar de un problema a un programa o itinerario de su solución, esto es, a una solución como una secuencia finita y determinística de pasos (Joyanes Aguilar, 2003). Esto implica la capacidad de definir y enunciar con claridad un problema; descomponerlo en subproblemas más pequeños y manejables y, describir una solución a lograr en un conjunto de pasos bien definido. La capacidad para analizar y dar solución a problemas, es una habilidad importante para todo ser humano, y en especial para los estudiantes (Polya, 1945).

El problema está en que para los niños de educación básica, trabajar y desarrollar algoritmos en ambientes computacionales, puede ser muy complicado ya que la sintaxis y semántica de las herramientas a utilizar para dicho propósito no son fáciles de comprender. Como se mencionó previamente, el lenguaje de programación animada Scratch, debido a sus características, soslaya esta dificultad y reviste con características de juego un trabajo netamente matemático-computacional produciendo un ambiente más propicio para el aprendizaje de los niños (De Guzmán, 1984). Así, de acuerdo con De Guzmán (1984), es posible afirmar que Scratch satisface ciertas condiciones didácticas para ser utilizado en la creación de ambientes tecnológicos para el desarrollo del pensamiento algorítmico en niños.

Diversos países reconocen al algoritmo como un elemento vital en el aprendizaje de los estudiantes. Por ejemplo, Inglaterra lo instala en 4 fases de su educación obligatoria, según indica Sophie Curtis (2014). Según Curtis (2014), los niños de 5 a 7 años (*Key Stage One*) aprenderán lo que son los algoritmos y los programas de ordenador, y que funcionan siguiendo unas instrucciones prefijadas. En el periodo de 7 a 11 años, diseñarán y escribirán programas, comprenderán el funcionamiento de las redes de computadores, y aplicarán el razonamiento lógico para detectar y corregir errores en los algoritmos. Entre los 11 y los 14 años se enseñará lógica booleana, entenderán cómo se convierte el pensamiento computacional en algoritmos, así como la estructura de los componentes hardware y software de los sistemas informáticos, la comunicación entre componentes y entre sistemas de computación. En el último periodo de educación obligatoria, de los 14 a los 16 años (*Key Stage 4*), el currículo está más abierto a la configuración particular por parte de cada centro. Otros países, tales como, Canadá, Israel, Japón, EEUU, Nueva Zelanda y España declaran el aprendizaje de algoritmos e informática educativa desde los primeros años de educación (PISA, 2014).

En Chile, los objetivos de aprendizaje transversales (OAT) para estudiantes de séptimo y octavo año de educación básica, y primero y segundo año de educación media (MINEDUC, 2014), declaran dentro de sus ámbitos el denominado “Tecnologías de Información y comunicación (TIC)” donde se propone un trabajo educativo que provea a todos los estudiantes de las herramientas para manejar en el llamado “Mundo digital” y desde él, poder desarrollarse con el uso competente y responsable de estas tecnologías. Dentro de los objetivos propuestos en esta dimensión se destacan por su relación con este trabajo: i) Utilizar TIC que resuelvan las necesidades de información, comunicación, expresión y creación dentro del entorno educativo y social inmediato; y ii) Utilizar aplicaciones para presentar, representar, analizar y modelar información y situaciones, comunicar ideas y argumentos, comprender y resolver problemas de manera eficiente y efectiva, aprovechando múltiples medios (texto, imagen, audio y video).

Lenguajes de Programación y Scratch

El proyecto Scratch, se inició en el año 2003, es un entorno de programación de computadores desarrollado por un grupo de investigadores del Lifelong Kindergarten del Laboratorio de Medios del MIT, bajo la dirección y liderazgo del Dr. Michael Resnick (Resnick et al., 2009; LEAD, 2012). Este proyecto cuenta con el apoyo de la National Science Foundation, Fundación Intel, Microsoft, Fundación MacArthur, Fundación LEGO, Fundación Code-to-Learn, Google, Dell, Fastly, Inversoft, y el consorcio de investigación del MIT Media Lab (Resnick et al., 2009; LEAD, 2012).

Tal como lo indican (Resnick et al., 2009; LEAD, 2012), el objetivo original de Scratch era desarrollar un enfoque de programación que atrajera a las personas, sin importar la edad, origen social, o educacional, al

desarrollo de soluciones algorítmicas sin las complejidades de sintaxis y semántica de los lenguajes de programación tradicional, esto es, hacer de Scratch un lenguaje para programar historias interactivas, juegos, animaciones y simulaciones fácil para todos sus usuarios, quienes pueden además compartir sus creaciones con otros. De esta forma, el objetivo principal de Scratch no es preparar a las personas para carreras profesionales o técnicas del área de programación, sino para nutrir una nueva generación de pensadores creativos y sistemáticos utilizando la programación para expresar sus ideas.

Desde el lanzamiento público en mayo de 2007, el sitio Web de Scratch (MIT, 2014) se ha convertido en una vibrante comunidad en línea, en donde sus miembros comparten, discuten y crean nuevas versiones de otros proyectos. Tal y como lo señala (MIT, 2014), Scratch ha sido llamado “el YouTube de los medios interactivos”, ya que cada día, usuarios de Scratch o Scratchers de todo el mundo suben más de 1.500 nuevos proyectos al sitio, cuyo código fuente se encuentra disponible libremente para ser compartido con los demás usuarios. La colección de proyectos que se encuentra en este sitio es tremendamente diversa, incluyendo juegos de video, boletines de noticias interactivas, simulaciones científicas, visitas virtuales, tarjetas de cumpleaños, concursos de baile de animación, y tutoriales interactivos, todos ellos programados en Scratch.

El público principal en el sitio está entre las edades de 8 y 16 años, aunque un grupo considerable de adultos también participa (MIT, 2014). Al programar con Scratch y compartir los proyectos interactivos, se pueden aprender importantes conceptos matemáticos y computacionales, así como la forma de pensar creativamente, razonar sistemáticamente y trabajar en colaboración: todas ellas, habilidades esenciales para el siglo 21. Ya en el año 2006, el documento “La Educación encierra un tesoro” (Delors, 2006) conocido como Informe Delors en honor al presidente de la comisión, el Francés Jacques Delors, menciona los 4 pilares de la educación del siglo XXI ; *aprender a conocer, aprender a hacer, aprender a vivir juntos y aprender a ser, todos ellos considerados en esta experiencia ya que el estudiante debe poner en práctica estos cuatro saberes para lograr dar respuesta a la interrogante y sobre todo generar el algoritmo que permite a Scratch cumplir su objetivo.*

Es importante señalar que, el nombre Scratch proviene de la técnica de scratching utilizada por los disc jockeys de hip-hop, quienes juegan con la música haciendo girar los discos de vinilo de ida y vuelta con las manos, mezclando los clips de música de forma creativa. En la programación de Scratch, la actividad es similar, mezclando gráficos, animaciones, fotos, música y sonido (Resnick et al., 2009; LEAD, 2012).

Cuando las computadoras personales se introdujeron por primera vez en la década de 1970 y 1980 (Resnick et al., 2009; LEAD, 2012), hubo entusiasmo inicial para enseñar a todos los niños a programar. En muchas escuelas se enseña a millones de estudiantes a escribir programas simples en lenguajes como Logo o Basic. En la década de 1980 se presenta a Logo como piedra angular para reflexionar sobre los enfoques a la educación y el aprendizaje. Aunque algunos niños y maestros fueron energizados y transformados por estas nuevas posibilidades, la mayoría de las escuelas pronto pasaron a otros usos de las computadoras. Desde entonces, las computadoras se han vuelto omnipresentes en la vida de los niños, pero pocos aprenden a programar. Hoy en día, la mayoría de la gente ve la programación de computadoras como una actividad estrecha, técnica, apropiada para sólo un pequeño segmento de la población.

De acuerdo al Learning through Engineering, Art, and Design Project (LEAD, 2012; MIT, 2014), los bloques de Scratch están conformados para ser encajados de acuerdo a su sentido semántico. Por ejemplo, las estructuras de control (como “siempre” y “repetir”) se representan con una letra “C”, es decir, con una abertura en uno de sus lados, para permitir colocar bloques en su interior, los cuales representan acciones a ser realizadas en cada iteración. Similarmente, los bloques condicionales (como “si” y “repetir-hasta”) tienen huecos hexagonales, lo que indica que se requiere una condición booleana para proceder a las acciones en su interior. Además, Scratch está diseñado para ser altamente interactivo: se pueden realizar cambios en un conjunto de bloques de acción mientras dichos bloques están en ejecución, por lo que es fácil de experimentar con nuevas acciones y sus resultados interactivamente. También permite crear hilos paralelos mediante la creación de múltiples conjuntos de bloques para uno o múltiples objetos.

ALGORITMOS Y EDUCACIÓN

Según MINEDUC (2014), las bases curriculares de la educación en Chile reconocen en los algoritmos uno de los elementos fundantes del pensamiento lógico matemático. Así mismo, según CURRÍCULUM (2014), para la organización curricular de los cursos de matemática en enseñanza básica, se define el eje Número y Operaciones de la siguiente forma: “Este eje abarca tanto el desarrollo del concepto de número como también la destreza en el cálculo mental y escrito. Una vez que los alumnos asimilan y construyen los conceptos básicos, con ayuda de metáforas y representaciones, aprenden los algoritmos de la adición, sustracción, multiplicación y división incluyendo el sistema posicional de escritura de números.”.

Claramente, el término *algoritmo* es reconocido como recurso para lograr que el estudiante construya conocimiento. En las mismas bases se propone como objetivo “*Resolver problemas rutinarios en contextos cotidianos, que incluyan dinero e involucren las cuatro operaciones*”. Desde un punto de vista práctico, estas bases recomiendan a los profesores impulsar que los estudiantes resuelvan todo tipo de problemas ya sea de manera individual o grupal, con el propósito de proveer el intercambio de estrategias o procedimientos de solución llegando a desplegar la habilidad de “resolución de problemas” producto del trabajo entre pares. De acuerdo al modelo pedagógico propuesto, se pide a los alumnos manejar material concreto o representaciones pictóricas y registrar el proceso en forma simbólica, aplicar estrategias de cálculo o el algoritmo correspondiente.

Fundamentos didácticos para la educación

El psicólogo y pedagogo Estadounidense Ausubel (1960) generó el concepto de Aprendizaje Significativo, donde él sostiene la idea de que los nuevos conocimientos se incorporan en forma sustantiva en la estructura cognitiva de los alumnos de escuela. Esto se logra cuando el estudiante relaciona los nuevos conocimientos con los anteriormente adquiridos; pero también es necesario que el alumno se interese por aprender lo que se le está mostrando. Es aquí donde las experiencias con Scratch que se describen en este documento cobran importancia al permitir al educando relacionar su experiencia con la generación de un procedimiento o algoritmo para cumplir sus objetivos, en este caso el de recoger un conjunto de pescados. De acuerdo a lo anterior, trabajar este programa producirá una retención más duradera de la información, facilitando la adquisición de los nuevos conocimientos de forma significativa, con claridad de la estructura cognitiva, que permitirá retener el nuevo conocimiento. Este aprendizaje debe estar relacionado con la asimilación de las actividades de aprendizaje por parte de los estudiantes, entendiendo el aprendizaje como un proceso personal que dependerá de los recursos cognitivos con que cuente.

Estas concepciones acerca del aprendizaje significativo se ven interpretadas por el proyecto LEAD (Resnick et al., 2009; LEAD, 2012), al afirmar que “comúnmente las personas aprenden mejor y disfrutan más cuando trabajan en proyectos que tienen un significado personal”. Es por esto, de acuerdo a (LEAD, 2012; MIT, 2014), que el desarrollo de Scratch pone una alta prioridad a dos criterios de diseño: (1) * Diversidad: Esto queda reflejado en el apoyo a muchos tipos diferentes de proyectos (cuentos, juegos, animaciones, simulaciones), por lo que las personas que tengan muy diversos intereses son capaces de trabajar en proyectos que les interesan; y (2) Personalización: Para las personas resulta bastante fácil personalizar sus proyectos de Scratch con la importación de fotos y clips de música, grabación de voces, y la creación de gráficos.

Por otra parte, el desarrollo del lenguaje de programación Scratch está estrechamente unido con el desarrollo del sitio web de Scratch (Resnick et al., 2009; LEAD, 2012; MIT, 2014). Para que Scratch tenga éxito, el lenguaje está vinculado a una comunidad donde los usuarios de Scratch pueden apoyar, colaborar y criticar unos a otros así como construir sobre el trabajo de los otros participantes de la comunidad. De esta forma, para fomentar el intercambio y la colaboración internacional, Scratch permite trabajar en diversos idiomas ya que una red global de voluntarios ha proporcionado traducciones para más de 40 idiomas. De esta forma, los niños de todo el mundo pueden compartir proyectos de Scratch con los demás, en donde cada uno puede ver los bloques de programación de Scratch en su propio idioma.

Tal y como lo indican (Resnick et al., 2009), un número creciente de escuelas en todo el mundo, e incluso algunas universidades (incluyendo Harvard y la Universidad de California, Berkeley), han incluido el uso de Scratch como un primer paso en la programación. Una pregunta natural es ¿Qué viene después? En los foros de discusión de Scratch, se encuentran debates sobre que lenguaje de programación se debe utilizar después de Scratch. En ellos también se encuentran peticiones para añadir características más avanzadas a Scratch (como herencia de objetos y estructuras de listas recursivas).

El Juego como estrategia de enseñanza

El juego constituye una situación que invita a la reflexión en el aprendizaje de la matemática; sin embargo, es importante que sea el profesor quien genere, participe y planifique juegos para los niños de educación primaria o básica en Chile para generar “situaciones de aprendizaje” con objetivos claramente definidos desde lo específico y didáctico mediante problemas desafiantes en el nivel adecuado. En este contexto, en los fundamentos de la teoría de situaciones didácticas, Brousseau señala: “El alumno aprende adaptándose a un medio que es factor de contradicciones, de dificultades, de desequilibrios, un poco como lo hace la sociedad humana. Este saber, fruto de la adaptación del alumno, se manifiesta por respuestas nuevas que son la prueba del aprendizaje.” (1986).

Es claro que, sobre los fundamentos didácticos existentes, el juego puede acercarse de manera importante a una situación didáctica en el sentido de Brousseau. En lo específico del juego como estrategia para el aprendizaje de la matemática, éste muestra dos aspectos diferenciados: uno formativo y otro informativo. El

primero está orientado a establecer una motivación y una experiencia directa en el estudiante por el uso del juego como un medio significativo para el aprendizaje de la matemática; despliega procesos motivacionales y de un hacer con sentido al ser actor principal en el proceso de aprender, el cual tendrá que poner en práctica cuando se enfrente a situaciones problemáticas. En cuanto al aspecto informativo agrega conocimientos, habilidades y destrezas en relación a los contenidos matemáticos, sobre todo en la planificación y ejecución de algoritmos que admitan el desarrollo de capacidades en los niños y niñas. Ambos aspectos son desarrollados en la experiencia con Scratch ya que este lenguaje de programación permite definir objetos gráficos con propiedades y acciones definidas, de manera de crear secuencia de interacción entre dichos objetos, esto es, definir algoritmos sobre dichos objetos, cuyos efectos son visibles sobre los objetos. Por este efecto de acción-reacción en los algoritmos definidos sobre objetos gráficos en Scratch, los estudiantes se motivan al ver directamente resultado de sus acciones sobre los objetos de soluciones algorítmicas, además de incorporar, muchas veces por descubrimiento, una serie de contenidos o conocimiento en su aprendizaje. Como se muestra en la siguiente sección, esto facilita el uso de elementos algorítmico básicos tales como estructuras condicionales y repetitivas, elementos que son parte de las situaciones diarias de todo ser humano.

El aprendizaje de la matemática obliga a introducir nuevos métodos pedagógicos y nuevas técnicas educativas que faciliten al niño aprender en los inicios de su vida escolar en forma lúdica y significativa. Al respecto Segarra (2002) señala dentro de los aspectos que deben introducirse para lograr aprendizaje matemático están los juegos de computadora donde el alumnado, jugando, pueda automatizar las operaciones elementales, conjuntamente con una Matemática recreativa y de pasatiempos.

APLICACIÓN Y RESULTADOS

Caso práctico con Scratch

En esta sección se describe el experimento realizado durante Mayo de 2014 en Linares con estudiantes de 1ero y 2do medio del colegio Diego Portales de Linares, y en Viña del Mar con estudiantes de 7mo básico de la escuela Ilusión N° 364 (Vidal et al., 2014). El experimento realizado fue el mismo para los estudiantes de ambos establecimientos. Los participantes del colegio Diego Portales fueron 12 alumnos entre séptimo básico y segundo medio, mientras que aquellos del colegio Ilusión fueron 15 alumnos de sexto básico. Para la realización exitosa del experimento, se realizó una introducción a Scratch y algunas acciones básicas de movimiento. Además, se destacó el uso de la interfaz de Scratch para la generación de escenarios con la participación de otros objetos.

Las preguntas planteadas a los estudiantes, tienen como contexto el escenario que se presenta en la figura 2, donde se aprecia a Scratch (gato protagonista del escenario del problema), en una esquina ubicada en la parte inferior de un camino triangular y 3 peces ubicados en la esquina superior de dicho camino. Estas preguntas se formularon en el momento inicial, antes de presentar Scratch como herramienta. La primera pregunta se refiere a la ruta más corta para que Scratch alcance los peces: ¿Cuál es el camino más corto que Scratch puede seguir para llegar a comer los peces? Las alternativas de respuesta fueron: 1. Caminar en forma horizontal y luego subir; 2. Caminar en forma diagonal y, 3. Ninguna de las anteriores. En un contexto de no interferencias en los caminos para lograr el objetivo de llegar a los peces, claramente, la respuesta correcta es la número 2. El rendimiento de los estudiantes para la primera pregunta se presenta en la tabla 1.

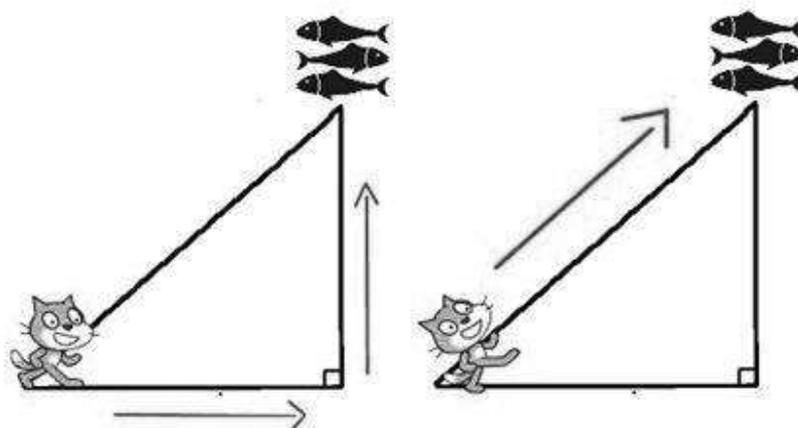


Fig. 2: Test de Scratch: Caminos posibles para alcanzar el objetivo.

Tabla 1: Rendimiento a Pregunta 1 en Test de Scratch.

<i>Establecimiento Educacional</i>	<i>Respuesta 1</i>	<i>Respuesta 2</i>	<i>Respuesta 3</i>
Escuela Ilusión, Viña del Mar.	2	7	3
Colegio Diego Portales, Linares.	1	12	3

Tras la pregunta inicial, en ambos establecimientos se procedió a enseñar Scratch como una herramienta animada y se identificó el personaje principal, el gato Scratch. Además, se explicó a los estudiantes cómo Scratch puede efectuar acciones de movimiento de una posición a otra, así como realizar cambios de orientación de manera de orientar su movimiento a diferentes direcciones. Entonces, se les explicó, como representar el escenario de la situación inicial para que los estudiantes resolvieran esta pregunta, lo cual fue un éxito en ambas realizaciones del test.

En la solución del problema inicial, una vez que Scratch alcanzaba la comida, se indicó a los estudiantes que imaginaran que Scratch comía un pez. Para hacer notar esta acción, se les enseñó el uso de sonido, luego se explicó una acción de control para realizar una pausa antes de realizar la siguiente acción de regresar a su origen para lo cual el gato debía reorientar su movimiento. Con esto, los estudiantes eran capaces de definir una secuencia finita de pasos para lograr un objetivo, definir una solución algorítmica. Cabe señalar que el triángulo de la figura 2, así como aquellos hechos por los estudiantes, no eran necesariamente rectángulos o equiláteros, así la medida del ángulo en el cual debía orientarse Scratch para alcanzar la comida no era necesariamente de 45 o 60 grados, ni era para todos la misma.

Tras la introducción a Scratch y sus acciones para que los estudiantes solucionaran la pregunta inicial, lo que en ambos casos duró menos de 30 minutos (en Valparaíso 20 y en Linares 25), se procedió a la siguiente pregunta del test relacionada con la cantidad de viajes que debía realizar Scratch para comerse todos los peces, asumiendo que en cada viaje Scratch sólo comía un pez y que los viajes se harían por el camino más corto. La pregunta 2 de este test fue: ¿Cómo el gato puede comerse toda la comida realizando la menor cantidad de acciones posible, si el gato puede comer sólo un pez en cada viaje que realiza desde su lugar de origen al que regresa después de comerlo? Las opciones de respuesta a esta pregunta, fueron las siguientes: 1. 3 viajes; 2. 5 viajes y, 3. Ninguna de las anteriores. Claramente la opción correcta es la 1., 3 viajes. Después de obtener las respuestas de los estudiantes, todas correctas tanto en Valparaíso como en Linares, se realizó la demostración de la solución, esta vez, con la ayuda directa de los estudiantes, quienes eran capaces de ir realizando su solución en sus computadores.

Después de responder a la segunda pregunta, tanto en Valparaíso como en Linares, se explicó a los estudiantes las consecuencias en el ahorro o gasto de espacio asociadas con la repetición de acciones. Se mostró el resultado en el caso que fueran 5 peces, y la solución de repetir los viajes de ida y regreso en 5 ocasiones, con lo cual el bloque de acciones a realizar por Scratch crece considerablemente. Entonces se mostró las ventajas de usar el bloque "repetir", que define un número de veces en que un conjunto de acciones en su interior son ejecutadas, es decir, para repetir una secuencia de acciones un número definido de veces. Este control de Scratch se denominada "repetir x". Entonces, para la solución de la segunda pregunta, se introduce la solución a la pregunta del test inicial en un bloque "repetir x", y se establece el número de repeticiones a 3. La figura 3 presenta la solución Scratch a la pregunta 2 del test con el uso de repetir, mientras la figura 4 presenta un pseudocódigo de la misma. Con esto, los niños conocieron lo que es un ciclo de repetición y cómo hacer más fácil el trabajo en Scratch. Como se aprecia en estas figuras, primero se asume que se está en presencia de un triángulo isósceles y se apunta en dirección de 45 grados. Luego, ya que se asume una distancia de 240 pasos y sólo se camina 80 pasos para luego descansar 1 segundo, es necesario repetir esta acción 3 veces para alcanzar 1 pez. Tras esto, se requiere regresar a su posición original por lo que simplemente retrocede, camino -80 pasos, para luego descansar. Todo esto, asumiendo que son 3 peces, es necesario repetirlo 3 veces. Nótese que no hay acción de sonido una vez alcanzado un pez. Además, una vez que concluye este conjunto de acciones, cuando se han alcanzado los 3 peces, se apunta en dirección 90 grados, esto es, en dirección del eje X.

Se realizó el test estadístico de McNemar para determinar si el aprendizaje de los estudiantes es estadísticamente significativo. Para el caso de Linares el χ^2 McNemar es de 4,9 y el χ^2 Crítico para un nivel de significancia de 0,05 es de 3,841. En el caso de Viña del Mar el χ^2 McNemar es de 5,1428. En ambos casos χ^2 McNemar > χ^2 Crítico por lo que podemos concluir que existe un aprendizaje después de la aplicación de Scratch. Cabe señalar que los experimentos tuvieron 60 minutos de duración, donde se incluye la introducción al uso de Scratch, y el desarrollo y solución de problemas a resolver, lo que demuestra la simplicidad de uso, y la motivación de los estudiantes por el uso de herramientas tecnológicas.

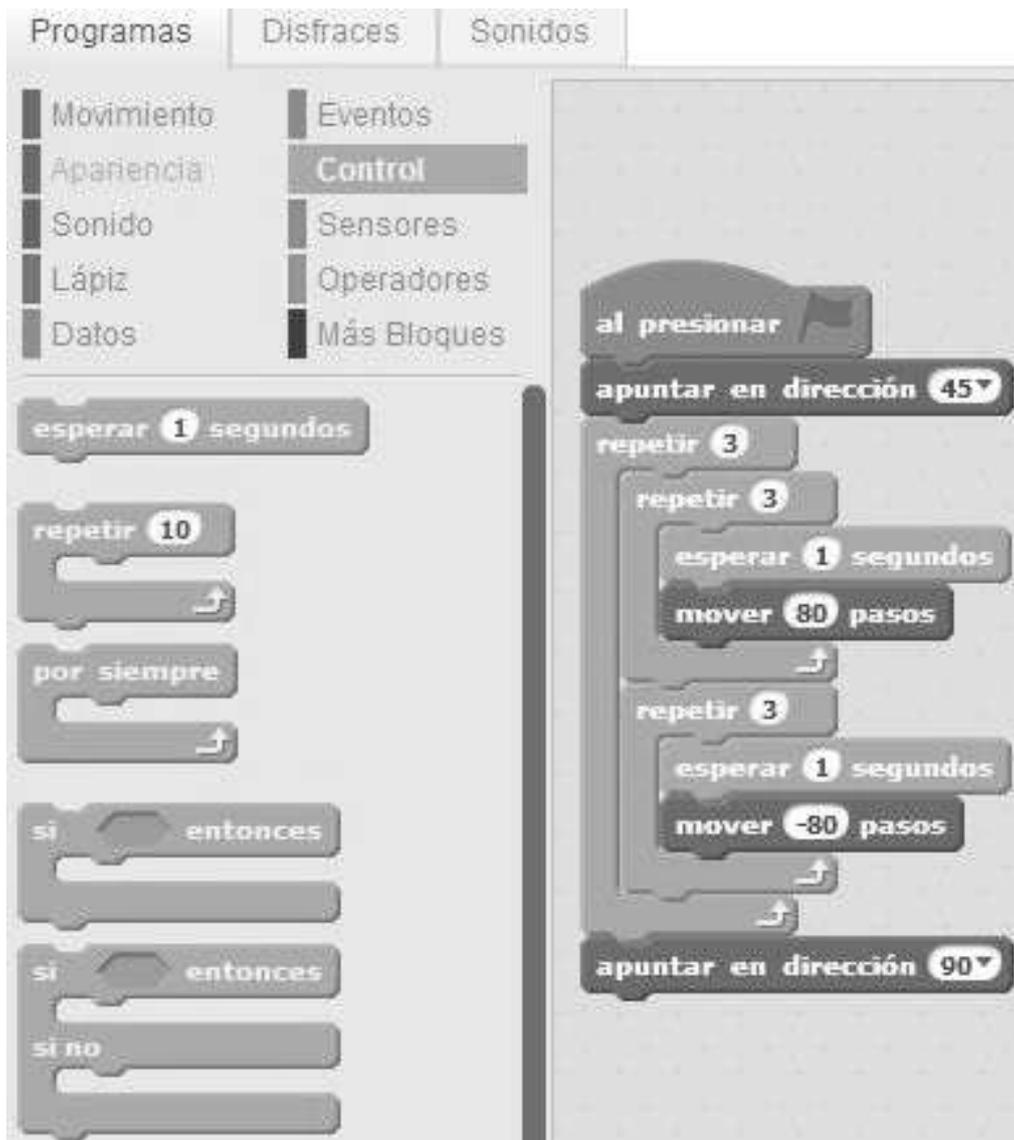


Fig. 3: Serier de Instrucciones Algorítmicas en Scratch.

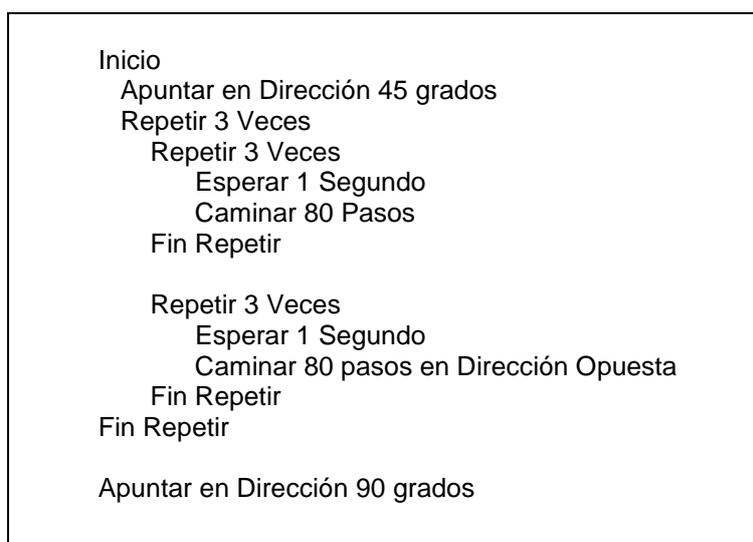


Fig. 4: Pseudocódigo de Solución Scratch de Figura 3.

CONCLUSIONES

La experiencia muestra que Scratch constituye una herramienta propicia para el desarrollo del pensamiento lógico y algorítmico para niños y estudiantes de Chile, y presenta un ambiente en el cual los estudiantes se motivan y participan en la propuesta de soluciones a las situaciones planteadas sin temor al error, posibilita el análisis de problemas y la propuesta, desarrollo y aplicación de soluciones lógicas y algorítmicas, las que se pueden probar y mejorar. Es decir, mediante pruebas de ensayo y error, los estudiantes pueden desarrollar y mejorar un pensamiento algorítmico. Cómo lo señala (Cooper et al., 2003), mediante el uso de objetos, es posible el desarrollo de razonamiento algorítmico, y Scratch permite trabajar directamente con las propiedades y acciones de objetos. De esta forma, Scratch es una herramienta adecuada para la enseñanza de algoritmos y programación. Justamente, en la realización de este experimento, fue posible comprobar que las alternativas de respuesta a una pregunta lógica, generan el razonamiento del estudiante sobre cuál es la opción correcta así cómo validar sus soluciones.

Es relevante destacar que, para la enseñanza de algoritmos también existen herramientas útiles como Pselnt (Pselnt, 2015), la cual es muy usada en cursos iniciales de programación ya que es permite definir elementos algorítmicos en Español. Primero, gracias a los objetos gráficos, segundo gracias a la nula posibilidad de errores sintácticos, salvo con la inclusión de variables, y por el efecto acción-reacción, este trabajo afirma que Scratch es más adecuado para el desarrollo de pensamiento algorítmico en niños.

Después de este trabajo, que representa una experiencia práctica de enseñanza, aprendizaje y uso de Scratch, para el desarrollo del pensamiento algorítmico, se puede afirmar que, a la luz de los resultados obtenidos, Scratch es una herramienta adecuada para la educación escolar por su versatilidad y posibilidades que ofrece a través del uso de recursos multimedia como sonido, gráfica y movimiento. Desde este punto de vista, permite el uso de técnicas de visualización para conjeturar y experimentar soluciones a problemas de diversa índole. En efecto, el contexto presentado en el experimento que se reporta en este artículo permite, con algunas modificaciones, plantear problemas de tipo geométrico relativos a triángulos u otras figuras planas.

REFERENCIAS

Ausubel, D.P, The use of advance organizers in the learning and retention of meaningful verbal material. *Journal of Educational Psychology*, 51, 267-272 (1960).

Acevedo, C., Arciniegas, J., García, X. y Perrinet, J., Proceso de Adaptación de una Aplicación de e-aprendizaje a t-aprendizaje, *Información Tecnológica*, 21(6), La Serena, Chile, Diciembre (2010).

Alice, Alice.org, Carnegie Mellon (en línea), 1999. <http://www.alice.org/index.php>. Acceso: 28 de Mayo (2014).

Brousseau, Guy, Fundamentos y métodos de la Didáctica de la Matemática, Universidad Nacional de Córdoba, Facultad de Matemática, Astronomía y Física, Serie B, Trabajos de Matemática, N° 19 (1986)

Cooper, S., Dann, W. y Pausch, R., Teaching Objects-first in Introductory Computer Science, *Actas de 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE'03, ACM, Reno, NV, USA*, 191-194 (2003).

Coutinho, S. C., Números Inteiros e Criptografia RSA, *Série de Computação e Matemática*, IMPA, Rio de Janeiro, Brasil, (2005).

CURRICULUM, Ministerio de Educación de Chile: Matemática – Organización Curricular (en línea), 2014. <http://www.curriculumlineamineduc.cl/605/w3-article-20852.html>. Acceso: 17 de Septiembre (2014).

Curtis, S., Teaching our children to code: a quite revolution, *The Telegraph* (en línea), 04 de Noviembre, 2013. <http://www.telegraph.co.uk/technology/news/10410036/Teaching-our-children-to-code-a-quiet-revolution.html>. Acceso: 17 de Septiembre (2014).

CVNE, Centro Virtual de Noticias de la educación, Ministerio de Educación de Colombia (en línea), 2006. <http://www.mineducacion.gov.co/cvn/1665/w3-article-241894.html>. Acceso: 28 de Mayo (2014).

De Guzmán, M., Juegos matemáticos en la enseñanza, *Actas de las IV Jornadas sobre aprendizaje y Enseñanza de las Matemáticas*, Santa Cruz de Tenerife, 10-14 Septiembre (1984).

Delors, J., *La Educación Encierra un Tesoro*, Santillana, Ediciones Unesco, (2006).

Joyanes Aguilar, L., *Fundamentos de Programación: Algoritmos y Estructuras de Datos*, Tercera Edición, McGraw-Hill Interamericana, Mayo (2003).

LEAD, *The Lead Project, Super Scratch Programming Adventure!: Learn to Program by Making Cool Games*, Primera Edición, No Starch Press (2012).

MINEDUC, Ministerio de Educación de Chile: Currículum Nacional (en línea), 2014. http://www.mineduc.cl/index5.php?id_portal=47. Acceso: 17 de Septiembre (2014).

MIT, Media Lab MIT, *Ayuda sobre Scratch - Acerca de Scratch* (en línea), 2003. <http://scratch.mit.edu/about/>. Acceso: 29 de Junio (2014).

Mora Sánchez, J. A., *Geometría Dinámica para el Análisis de Obras de Arte*, Unión: Revista Iberoamericana de Educación Matemática, ISSN-e 1815-0640, 9, 83-99 (2007).

Pacheco Vera, F., *Formas de pensar y su utilización en el ámbito empresarial y profesional, Experiencias y Propuestas en la Construcción del Estilo Pedagógico en Diseño y Comunicación*, XV Jornadas de Reflexión Académica 2007, ISSN 1668-1673, Universidad de Palermo, Buenos Aires, Argentina, Febrero (2007).

PISA, Programa para la Evaluación Internacional de los Alumnos (en línea), 2012. http://iaqse.caib.es/documents/pisa2012/PISA_2012_linea_volumenI.pdf. Acceso: 17 de Septiembre (2014).

Pólya, G., *How to Solve It*, ISBN 0-691-08097-6, Princeton, USA (1945).

PseInt, *..una invitación a entrar en el mundo de la programación...* (en línea), 2003. <http://pseint.sourceforge.net/>. Acceso: 31 de Enero (2015).

RAE, Real Academia Española, *Diccionario de la lengua Española - Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema* (en línea), 2014. <http://lema.rae.es/drae/?val=algoritmo>. Acceso: 28 de Mayo (2014).

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

Scratch, *Scratch – Imagina, Programa, Comparte*, MIT Media Lab (en línea), 2007. <http://scratch.mit.edu/>. Acceso: 28 de Mayo (2014).

Scratch, *Scratch – Imagina, Programa, Comparte*, MIT Media Lab (en línea) 2007. <http://scratch.mit.edu/>. Acceso: 10 de Mayo (2014).

Segarra, L., *Juego y Matemáticas, La Resolución de Problemas en Matemáticas*, Caracas: Laboratorio Educativo (2002).

Vidal, C., Zamora, V. y Conejeros, P., *Experiencia práctica con Scratch para el desarrollo del pensamiento algorítmico en Chile*, INFONOR, Universidad Arturo Prat, Iquique, Chile, Julio (2014).