

Desarrollo del Pensamiento Computacional con Scratch

Rosa Barrera Capot

Universidad de Santiago de Chile
Las Sophoras 175
Departamento de Matemática y Ciencia de la
Computación
56-2-27182 005
rosa.barrera@usach.cl

Rosa Montaña Espinoza

Universidad de Santiago de Chile
Las Sophoras 175
Departamento de Matemática y Ciencia de la
Computación
56-2-27182 040
rosa.montano@usach.cl

ABSTRACT

This article presents a strategy to develop the computational thinking, using interactive activities in Scratch. This work is in the framework of a project of teaching innovation of the University of Santiago of Chile.

Designed eleven working sessions that will go from the level of recognition and data management to automation and simulation of problems in daily life of medium and high complexity problems. The sessions are hierarchically graduated and in each one of them we add skills to enable them to go reaching the highest levels of computational thinking.

The results show that interactive activities with Scratch resources achieve the aim motivating the user so that at the end of the sessions are achieved the highest levels of computational thinking. Finally, the creativity of the student thrives in the last sessions, where activities are more open and interactive than the initials.

RESUMEN

Este artículo presenta una estrategia para desarrollar el pensamiento computacional, utilizando actividades interactivas en Scratch. Este trabajo se enmarca en un proyecto de innovación docente de la Universidad de Santiago de Chile.

Se diseñaron once sesiones de trabajo, que permitirán ir desde el nivel de reconocimiento y manejo de datos hasta automatización y simulación de problemas de la vida diaria, de complejidad media y alta. Las sesiones están graduadas jerárquicamente y en cada una de ellas se agregan habilidades que permitan ir alcanzando los niveles más altos del pensamiento computacional.

Los resultados muestran que los recursos de actividades interactivos con Scratch, logran el objetivo planteado, motivando al usuario, y alcanzando al término de la experiencia niveles más

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '10, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

altos del pensamiento computacional. Finalmente, la creatividad del estudiante prospera en las últimas sesiones, donde las actividades son más abiertas e interactivas que las iniciales.

Categories and Subject Descriptors

New trends in learning with digital technology

General Terms

Performance, Design,

Keywords

Pensamiento Computacional, game-based learning. Interactividad

1. INTRODUCCIÓN

El término pensamiento computacional fue utilizado por primera vez, en 1996, por Seymour Papert [1]. El que se define como un método de resolución de problemas y es llamado así por su extenso uso en las técnicas de las ciencias de la computación [5]. Además, son los procesos de pensamiento involucrados en la formulación de problemas y sus soluciones, de modo que tales soluciones se representan en una forma que pueda ejecutar eficazmente un agente de procesamiento de información [18].

El pensamiento computacional es un enfoque para resolver un determinado problema que empodera la integración de tecnologías digitales con ideas humanas [4]. Refuerza el énfasis en creatividad, razonamiento o pensamiento crítico al tiempo que realza formas de organizar el problema de manera que el computador pueda ayudar.

El pensamiento computacional refuerza los estándares educativos en todas las asignaturas para acrecentar la habilidad del estudiante de solucionar problemas y así desarrollar pensamiento de orden superior [4]. El estudiante logra avances cuando usa algoritmos para resolver problemas y mejora la solución de estos con la computación; cuando analiza textos y construye comunicaciones complejas; cuando analiza grandes grupos de datos e identifica patrones a medida que adelanta investigaciones científicas.

Este pensamiento está relacionado con otros pensamientos tales como: pensamiento lógico, pensamiento analítico, pensamiento algorítmico, pensamiento abstracto, pensamiento divergente y

pensamiento crítico [8]. Estos pensamientos son importantes ya que cada uno de estos abarca una rama en la cual se involucra el pensamiento computacional.

El pensamiento computacional [4], tal como se definió anteriormente, es un proceso de solución de problemas que incluye, principalmente, las siguientes características:

- Formular problemas de manera que permitan usar computadores y otras herramientas para solucionarlos.
- Organizar datos de manera lógica y analizarlos
- Representar datos mediante abstracciones, como modelos y simulaciones
- Automatizar soluciones mediante pensamiento algorítmico (una serie de pasos ordenados)
- Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva
- Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.

Las características expuestas serán la base de las actividades que contiene el recurso. Algunos de los principales procesos identificados por Computer Science Teachers Associations (CSTA) and International Society for Technology in Education (ISTE) [4], corresponden a:

- Recopilar Datos: El proceso de reunir la información apropiada.
- Analizar Datos: Encontrarle sentido a los datos, hallar o establecer patrones y sacar conclusiones
- Representar Datos: Representar y organizar los datos en gráficas, cuadros, palabras o imágenes apropiadas
- Descomponer Problemas: Dividir una tarea en partes pequeñas y manejables
- Abstractar: Reducir la complejidad para definir o establecer la idea principal.
- Algoritmos y Procedimientos: Serie de pasos ordenados que se siguen para resolver un problema o lograr un objetivo
- Automatización: Hacer que computadoras o máquinas realicen tareas tediosas o repetitivas
- Simulación: Representar o modelar un proceso. La simulación involucra también realizar experimentos usando modelos.
- Paralelismo: Organizar los recursos para que simultáneamente realicen tareas con el fin de alcanzar una meta u objetivo común

Estos procesos se utilizan en la generación de algoritmos para la solución de problemas de diferente índole, tamaño y contexto. Esta tarea es considerada abstracta, por lo que algunos investigadores han planteado desarrollarla de forma lúdica, por medio de juegos que motiven su aprendizaje y así convertirla en una actividad accesible.

El juego es una actividad libre y voluntaria, regida por el principio de placer y disfrute personal; estructurada por un comienzo, desarrollo y fin, sujeta a límites espacio-temporales

propios y a reglas que condicionan dicho desarrollo, pero no así el final, que dependerá de la iniciativa de los/as participantes manteniendo el principio de la sorpresa y emoción, gracias a la incertidumbre del resultado final [13, 15]. En las actividades a desarrollar, la estructura está determinada por las características mencionadas anteriormente.

Un juego educativo es un juego que tiene un objetivo educativo implícito o explícito para que se aprenda algo específico. Un objetivo que explícitamente programa el educador con un fin educativo, y está pensado para que se aprenda algo concreto de forma lúdica [11,12]. Los juegos a utilizar serán de ordenador, ya que generan un vínculo directo con el área en que se va a desarrollar el juego.

Los juegos de ordenador educativos pueden llegar a ser una herramienta muy eficaz para enseñar cosas concretas a personas de todas las edades [14]. En lo que respecta al poder individual, los juegos desenvuelven el lenguaje, despiertan el ingenio, desarrollan el espíritu de observación, afirma la voluntad y perfeccionan la paciencia. También favorecen la agudeza visual, táctil y auditiva; aligeran la noción del tiempo, del espacio; dan soltura, elegancia y agilidad del cuerpo.

Según Coopman las personas aprenden mejor cuando están entretenidos, cuando se puede utilizar la creatividad para trabajar hacia objetivos complejos, cuando la lección incorpora tanto el pensamiento y la emoción, y cuando se pueden observar las consecuencias de las acciones, en este contexto los juegos educativos cumplen con un rol relevante, ya que generan y potencian habilidades en diferentes materias o áreas del conocimiento [6,8,10].

Debido a los potenciales beneficios que los juegos pueden brindar al aprendizaje, estos se están considerando cada vez más como herramienta para que los estudiantes logren entender de una forma entretenida, rápida y clara [7,16,17].

En la actualidad existen diversas plataformas computacionales donde crear juegos y/o actividades, además algunas de estas permiten desarrollar el aprendizaje de la programación o generar el pensamiento computacional. Una de ellas corresponde a Scratch que fue desarrollado por el “Lifelong Kindergarten group” en el Media Lab del Instituto Tecnológico de Massachusetts como un lenguaje de programación visual para niños de 6 años en adelante. Scratch es gratuito y puede correr en plataformas Windows, Mac y Linux [3].

Por último cabe destacar que el recurso a desarrollar se diferencie del resto, en que se centra en la enseñanza del pensamiento computacional, no en un lenguaje de programación ni en el pensamiento ingenieril. Además, involucra la utilización de diferentes recursos, tales como Scratch, ya que será en esta donde se desarrollarán las diferentes actividades que generan este pensamiento.

2. DISEÑO DE ESTRATEGIA

Desarrollar recursos, para generar habilidades lógicas de programación utilizando Scratch, estos recursos contienen varias actividades donde cada una tiene un objetivo diferente y están centradas en diferentes niveles del pensamiento computacional. Estas actividades permiten generar reportes estadísticos de los estudiantes, para así visualizar el progreso de cada uno de ellos, además de ser jerárquicas en los niveles del pensamiento computacional.

La finalidad de los recursos es que estos puedan ser utilizados por alguien con competencias en el área de la computación y/o educación para preparar y dar clases a estudiantes interesados en el tema, los recursos son actividades en Scratch a realizar por los estudiantes y estas serán guiadas por el profesor a cargo, o bien ser auto guiadas por siguiendo las instrucciones del diseño para cada una de las actividades, si corresponden a niños de enseñanza básica se sugiere que sean guiadas por un tutor..

Estos recursos están diseñados en 11 sesiones (las sesiones se pueden ver en www.algoritmos.cedetec.cl), cada una de estas sesiones tiene una duración aproximada desde una 1 hora y 30 minutos a 2 horas, por lo que estas podrán ser parte de algún curso de verano, taller en algún colegio, etc. Cada sesión tiene un objetivo de lo que debe aprender el estudiante al finalizar la sesión. Al término se pretende que el usuario haya alcanzado un nivel alto del pensamiento computacional.

La siguiente tabla presenta cada una de las sesiones del curso y los procesos del pensamiento computacional que están presentes en cada una.

Tabla1: Descripción de las sesiones

Nº	Título	Objetivo	Pensamiento Computacional
1	Conociendo Scratch	Conocer entorno de trabajo y características del software, por medio de juegos y ejemplos en Scratch	Recopilación de Datos
2	Uso de variables simples para generación de animaciones.	Identificar tipos de variables simples que permiten movimientos, en soluciones a problemas simples que se puedan representar computacionalmente.	Recopilación de Datos, Análisis de Datos y Representación de Datos.
3	Creación de animaciones de Scratch en la generación de simulaciones de situaciones de la vida diaria	Realizar simulaciones de situaciones de la vida diaria.	Recopilación de Datos, Análisis de Datos y Simulación

Nº	Título	Objetivo	Pensamiento Computacional
4	Programar en Scratch soluciones a problemas simples matemáticos	Realizar algoritmos y procedimientos utilizando procesos lógicos	Recopilación de Datos, Análisis de Datos, Algoritmos y Procedimientos y Automatización de Procesos.
5	Crear una animación basada en una historia, generando así una secuencia lógica.	Organizar información para dividir una tarea en partes pequeñas y manejables para generar una secuencia lógica que resuelva el problema.	Recopilación de Datos, Análisis de Datos, Descomposición de Problemas, Algoritmos y Procedimientos
6	Crear una solución para representar el desplazamiento de una persona.	Reunir y analizar la información para generar desplazamiento de persona y/o objetos.	Recopilación de Datos y Análisis de Datos, Realización de Algoritmos y Procedimientos, Simulaciones de Situaciones
7	Crear juegos Scratch usando interacciones entre distintos elementos.	Realizar un juego en Scratch que integre los elementos vistos en las sesiones anteriores	Recopilación y Análisis de Datos, Descomposición de Problemas.
8	Crear un laberinto interactivo en Scratch	Crear juegos que contemplen en su realización la capacidad de descomposición de problemas, realización de algoritmos y procedimientos.	Descomposición de Problemas, Realización de Algoritmos y Procedimientos
9, 10 y 11	Crear un juego en Scratch a elección del estudiante en el que se aplique todo lo enseñado en el curso.	Crear juegos que contemplen en su realización la capacidad de recopilación y análisis de información, descomposición de problemas, realización de algoritmos y procedimientos y automatizar procesos.	Recopilación y análisis de datos, descomposición de problemas, realización de algoritmos y procedimientos y automatizar procesos.

Transversalmente en todas las sesiones están presentes los elementos más básicos del pensamiento computacional, pero a

medida que estas avanzan se focalizan en la profundización de procedimientos y automatización

Por último, se aplicó un test diagnóstico y un test al término del proceso [9] que identifica el nivel de pensamiento computacional que tiene una persona. Los niveles de pensamiento computacional considerados van desde recolección de información hasta automatización.

3. RESULTADOS

El material, se aplicó a dos muestras diferentes. La primera es un taller complementario a segundo ciclo de enseñanza básica, donde un grupo de 19 alumnos (9 de séptimo y 10 de octavo) de un colegio particular subvencionado de hombres, tomaron en forma voluntaria el taller durante un periodo de dos meses, con dos sesiones a la semana de dos horas pedagógicas, las actividades 10 y 11 se trabajaron en más de una sesión de trabajo, ya que les resulto complejo terminar la actividad en la sesión programada.

La segunda muestra corresponde a 53 estudiantes que ingresan a la universidad, en particular a la Carrera Licenciatura en Ciencia de la Computación. En la asignatura de Computación I, se genera una unidad 0, donde los estudiantes tienen que desarrollar las actividades propuestas, esta unidad se ve en 4 sesiones de dos horas pedagógicas, en un comienzo se avanza en más de una actividad por sesión, y se enfatiza en las cuatro últimas que apuntan a niveles más profundos del pensamiento computacional.

Tabla2: Descripción de los resultados.

Tipo de Estudiante	Resultado
Segundo Ciclo de Enseñanza Básica	1. 100% logra la realización de las actividades. 2. 20% logra automatizar los procesos. Aunque todos realizan la última actividad, sólo un grupo minoritario incorpora elementos nuevos y realizan interacciones nuevas a las vista durante las sesiones de trabajo.
Primer nivel carrera Licenciatura en Ciencia de la Computación	1. Las primeras actividades se realizan en promedio de una hora, demorando menos del tiempo esperado, por lo que se realizan varias actividades en una sesión de trabajo. 3. 100% logra la realización de las actividades. 2. 85% logra automatizar los procesos. La gran mayoría de los estudiantes incorpora elementos nuevos, interacciones no vistas y complejidad superior a la esperada.

En términos generales, en ambas experiencias, al término de la onceava sesión, el usuario cuenta con habilidades de orden superior respecto al pensamiento computacional. Sin embargo, los estudiantes de enseñanza básica necesitan más tiempo para procesar la información. Un aspecto a destacar, es que todos

logran la realización actividades, la tabla N° 2 muestra algunas especificaciones de los resultados obtenidos.

4. CONCLUSIONES

En este trabajo se diseñó una estrategia para desarrollar el pensamiento computacional en personas, de los medios y formas revisadas, lo más recomendado fue por medio de juegos interactivos, pero más que jugar también creando estos juegos. La revisión de plataformas que permiten realizar juegos de forma lúdica nos llevó a Scratch, herramienta altamente utilizada hoy en día en los procesos de enseñanza interactiva, está permite la generación de contenidos mediante una programación basada en bloques y en base a las funcionalidades que ofrece, se planifican los recursos y se desarrollan las actividades y se triangula con las componentes del pensamiento computacional

Los resultados obtenidos son alentadores y nos indican que estamos en el camino correcto, ya que se logra el objetivo propuesto y además las actividades interactivas que dejan una ventana abierta a la creatividad del aprendiz. El estudiante puede llegar a tener un pensamiento computacional avanzado con las actividades diseñadas, en forma lúdica y entretenida.

Quisiéramos destacar la creatividad que tienen los estudiantes y como esta emerge en forma natural en las últimas sesiones, el concepto de hacer un juego para un estudiante es un desafío y las reglas que ellos se imponen superan las expectativas de lo que se pretende. En este contexto, el juego una vez realizado es socializado y puesto a prueba por sus compañeros, resulta gratificante para el estudiante que su producto sea algo tangible y que otra persona lo quiera utilizar. También presenta un desafío su mejora para hacerlo lo más atractivo.

En suma, las actividades interactivas para generar pensamiento computacional de alto nivel son una herramienta efectiva que ayuda a fortalecer los procesos lógicos que permiten la modelación correcta de solución de problemas, además de fomentar habilidades como la creatividad. Hay que seguir incursionando para establecer el efecto que provoca transversalmente en la resolución de problemas, en un ámbito más científico.

5. REFERENCIAS

- [1] Bers, M. U. (2010). El programa de robótica TangibleK. Pensamiento computacional aplicado para niños pequeños. Investigación y Práctica de la Niñez Temprana.
- [2] Brennan, K. and Resnick, M. (2012) Nuevos Marcos de Referencia para Estudiar y Evaluar el Desarrollo Del Pensamiento Computacional, encuentro anual de la “American Educational Research Association”, AERA 2012, Vancouver, BC, Canada.
- [3] Cabral, I. (2012). Enseñando a Niños a Programar. SG, 50-52.
- [4] Computer Science Teachers Associations (CSTA) and International Society for Technology in Education (ISTE) (2011). Computational thinking teacher resources(2nd ed.) Recuperado el 15 de Mayo de 2015, de

- http://www.csta.acm.org/Curriculum/sub/CurrFiles/472.11C_TTeacherResources_2ed-SP-vF.pdf
- [5] Educadores Digitales. (2014). Educadores Digitales. Recuperado el 6 de Abril de 2014, de <http://www.educadoresdigitales.org/2012/03/pensamiento-computacional-recursos.html>
- [6] Figueroa, S. (16 de Abril de 2013). eldiario.es. Recuperado el 22 de Abril de 2014, de http://www.eldiario.es/turing/ninos-sabian-programar_0_122487936.html
- [7] Gerardo, H. (8 de Febrero de 2009). educar. Recuperado el 22 de Abril de 2013, de <http://portal.educar.org/foros/importancia-del-juego-en-la-educacion>
- [8] 8Leeuwen, S. V. (17 de Octubre de 2012). ABC.es. Recuperado el 23 de Abril de 2014, de <http://www.abc.es/videojuegos/nintendo-3ds/item/1492-uno-de-cada-tres-profesores-utilizan-videojuegos-de-apoyo-en-el-aula.html>
- [9] Moreno, J. (2015). Dr. Scratch, análisis de proyectos Scratch para medir el desarrollo del Pensamiento Computacional y mejorar las habilidades de programación. Seminario eMadrid sobre Pensamiento Computacional, recuperado el 7 de septiembre de 2015, de <http://www.emadridnet.org/es/dr-scratch-analisis-de-proyectos-scratch-para-medir-el-desarrollo-del-pensamiento-computacional-y-mejorar-las-habilidades-de-programacion>
- [10] Montenegro, L. (29 de Noviembre de 2010). Lengua y Tecnología Educativa. Recuperado el 23 de Abril de 2015, de <http://redpex.0.portafolioseducativos.com/lenguaytecnologia/2010/11/29/uso-de-video-juegos-en-clases-de-lengua/>
- [11] Ramos, A. G. (Enero de 2008). Una manera de romper estereotipos sexistas en el nivel inicial. Mexico D.F.
- [12] reEduca.com. (9 de Diciembre de 2010). reEduca.com. Recuperado el 15 de Mayo de 2013, de <http://www.reeduca.com/juego-educativo-concepto.aspx>
- [13] Revista de Educación a Distancia (2015). Número monográfico sobre "Pensamiento computacional". Publicación en línea. Murcia (España). Año XIV. Número 46. 15 de Septiembre de 2015. Recuperado el 2 de octubre de 2015 de <http://www.um.es/ead/red/46/>.
- [14] Rodríguez, L. (2004). Juegos Educativos Para Alumnos de Educación Primaria. Recuperado el 10 de noviembre de 2014, de <http://www.diegoweb.net/juegos%20educativos/>
- [15] Suárez, N. d. (2010). scribd.com. Recuperado el 29 de Mayo de 2014, de <http://es.scribd.com/doc/7128885/Teoria-Del-Juego>
- [16] Torres, C. (5 de Febrero de 2011). formared. Recuperado el 22 de Abril de 2014, de <http://formared.blogspot.com/2011/02/ventajas-y-desventajas-de-los.html>
- [17] Villoslada, A. (2013). todoPAPÁS. Recuperado el 23 de Abril de 2014, de <http://www.todopapas.com/ninos/educacion/videojuegos-educativos-una-herramienta-para-aprender-3859>
- [18] Wing, J. M. (2011). School of Computer Science. Recuperado el 7 de Mayo de 2015, de <http://link.cs.cmu.edu/article.php?a=600>