

Scratch y Necesidades Educativas Especiales: Programación para todos

Scratch in Special Education: Programming for All

Carmen López-Escribano
Universidad Complutense de Madrid
carmenle@edu.ucm.es

Rafael Sánchez-Montoya
Universidad de Cádiz
rafael.sanchez.montoya@uca.es

Resumen

Este trabajo tiene como objetivo principal animar a los maestros y educadores a utilizar el lenguaje de programación Scratch en sus clases con alumnos con necesidades educativas especiales. Scratch es un lenguaje visual de fácil uso y de libre distribución que favorece un método de aprendizaje activo y constructivo. Para que los maestros se animen a utilizar este programa exponemos experiencias de uso del programa con niños con diferentes necesidades y estilos de aprendizaje y ofrecemos una lista de recursos para hacer el programa accesible. También proponemos diferentes usos del programa, como es crear proyectos que sirvan para la rehabilitación de dificultades de aprendizaje. Las experiencias de utilizar Scratch con estudiantes con NEE, realizadas hasta la fecha, han sido siempre positivas y motivadoras.

Abstract

The main aim of this paper is to encourage teachers and educators to use Scratch in their classrooms with special education students. Scratch is a visual, easy to use, and free programming language which promotes active and constructive learning. To encourage teachers to use this program we offer other teachers' experiences with different abilities and learning style students and a list of assistive technology resources to make Scratch accessible to all. We also suggest other ways to use this program, as to make projects for learning disabilities rehabilitation. The experiences of using Scratch with special education students, presented below, are all positive and encouraging.

Palabras clave: Programación; Educación Especial; Tecnología Asistiva; Rampas Digitales

Key words: Programming; Special Education; Assistive Technology

Introducción

En general los estudiantes piensan que programar es difícil, de hecho, los cursos de introducción a la programación tienen una alta tasa de abandono y normalmente los alumnos no aprenden a programar bien. Muchas personas ven la programación como una actividad técnica destinada sólo a un pequeño grupo de la población (Radenski, 2006). El alto nivel de abstracción y la complejidad de los conceptos que deben ser aprendidos para programar es un impedimento serio para muchas personas. Para la mayoría programar es una actividad compleja y aburrida, destinada, a lo que se denomina en lengua inglesa “*computer geeks*” o en nuestra lengua “frikis de la informática”.

No es de extrañar, por tanto, que algunos lenguajes de programación como Logo, y otros similares, destinados al mundo educativo no han sido utilizados de modo extensivo, ni han perdurado en el tiempo. Resnick et al. (2009) exponen las siguientes razones para que esto haya sucedido:

- Las primeras lenguas de programación eran muy difíciles de usar y muchos niños no podían aprender la sintaxis de la programación.
- La programación era introducida con actividades como, por ejemplo, generar una lista de números primos, que no resultaban interesantes, ni atractivas a los jóvenes, ya que no estaban conectadas con sus intereses y experiencias.
- La programación se enseñaba en contextos donde era difícil obtener ayuda cuando algo no funcionaba.

El diseño del lenguaje de programación Scratch ha pretendido superar estas limitaciones: es un lenguaje visual y no hay que escribir líneas de programación, por tanto se evitan los errores al teclear; se pueden realizar todo tipo de proyectos y actividades personalizadas utilizando recursos multimedia; la web de Scratch permite compartir los proyectos realizados y obtener asesoramiento de otras personas. Más adelante desarrollamos con más detalle todos estos puntos.

Si, como comentamos más arriba, aprender a programar es considerado una tarea ardua, en principio, no parecería factible la posibilidad de proponer actividades de programación para alumnos con Necesidades Educativas Especiales (en adelante NEE). En los párrafos siguientes reflexionaremos sobre este tema. En primer lugar presentaremos Scratch un lenguaje de programación visual de libre distribución. Después reflexionaremos sobre el uso de Scratch con alumnos con NEE y cómo hacer que Scratch sea accesible para todos, comentaremos también algunas experiencias de su utilización con estudiantes con NEE. A continuación, revisaremos un programa diseñado con Scratch para mejorar aspectos de la lectoescritura y finalmente presentaremos las conclusiones.

Scratch

Scratch es un programa de libre distribución, disponible en 50 lenguas diferentes, que se fundamenta en las ideas de aprendizaje constructivistas de Logo y que se puede bajar de la página Web:

http://info.scratch.mit.edu/es/Scratch_1.4_Download

Scratch es un entorno de programación visual que permite a los usuarios crear proyectos multimedia interactivos. Un gran número de personas ha creado una amplia variedad de proyectos, utilizando Scratch, como videos musicales, presentaciones, juegos de ordenador y otro tipo de animaciones.

Los proyectos de Scratch contienen “media” y “scripts”. Las imágenes y los sonidos pueden ser importados o creados en Scratch utilizando herramientas construidas

en el propio programa. La programación se realiza ensamblando bloques de comandos, de diferentes colores, para controlar objetos gráficos en 2-D llamados “duendecillos” (sprites) que se mueven en un fondo llamado “escenario” (stage). Los proyectos creados con Scratch pueden salvarse o se pueden compartir en la página Web de Scratch. (Maloney, Resnick, Rusk, Silverman & Eastmong, 2010).

Un punto fuerte de Scratch es su página Web scratch.mit.edu, se trata de una Web social, donde los usuarios que se registran pueden compartir sus proyectos, comentar proyectos realizados por otros, formar grupos con intereses comunes, agrupar proyectos en galerías y otras muchas cosas más. Esta página Web es bastante clara y explicativa, en la parte superior, encontramos los links “inicio”, “proyectos”, “galerías”, “soporte”, “foros”, “acerca de”, “mis cosas” e “idioma”. En el link “soporte” están las guías y documentos que nos muestran de modo claro y sencillo cómo utilizar Scratch.

Papert defendía que las lenguas de programación debían tener un “suelo bajo” (fácil empezar a programar) y un “techo alto” (oportunidades de crear proyectos más complejos con el tiempo). Además, los creadores de Scratch, Resnick et al. (2009) añaden, que los lenguajes de programación necesitan “amplias paredes” (capacidad de englobar diferentes tipos de proyectos para que personas con intereses y estilos muy diversos se animen a programar). Para conseguir que estos objetivos sean posible, los creadores de Scratch (Resnick et al. 2009) han introducido tres principios o características básicas en el diseño de este lenguaje de programación. Estos principios son: que la lengua de programación sea lúdica, significativa y social. A continuación explicamos brevemente estos tres principios imbuidos en el diseño de Scratch.

El lenguaje de programación debe ser lúdico. La idea es que la lengua de programación facilite el juego y que se puedan probar, con facilidad, diferentes opciones. Los autores comparan la forma de programar en Scratch con la forma de construir objetos con Lego. Los bloques o ladrillos de construcción de Lego tienen unos conectores que sugieren como van unidos unos con otros y es fácil jugar con ellos y empezar a construir objetos. En este caso el “suelo es bajo” y la experiencia es lúdica. De modo similar, Scratch tiene unos “bloques de programación” de diferentes colores, con conectores que permiten que se puedan encajar unos con otros. El objetivo es que los niños puedan jugar con ellos desde el principio y probar a construir sencillos programas (Resnick et al. 2009).

La experiencia al utilizar el lenguaje de programación debe ser significativa. Sabemos que uno de los principios del aprendizaje es que las personas aprendemos más y disfrutamos más, cuando trabajamos en proyectos personalmente significativos. De este modo en el diseño de Scratch, sus creadores, han dado prioridad a dos criterios del diseño: diversidad (que pueda soportar diferentes tipos de proyectos: historias, juegos, animaciones, simulaciones) y personalización (que los proyectos se puedan personalizar importando fotos, voces, gráficos, etc.)

El uso de la lengua de programación debe propiciar la interacción social. El desarrollo de Scratch va muy unido al desarrollo de su página Web. Para que Scratch tenga éxito necesita que una gran comunidad de personas comparta, apoye, critique, colabore y pueda construir sobre el trabajo de otros. Así el concepto de “compartir” está construido en el entorno de interfaz de usuario de Scratch. Haciendo un clic sobre

“compartir” los proyectos suben a la página Web de Scratch para ser compartidos. Otras personas apoyaran, criticaran y construirán sobre los proyectos de otros, el objetivo es que finalmente resulte una experiencia de aprendizaje interactiva y enriquecedora para todos (Resnick et al. 2009).

¿Por qué Scratch para las personas con NEE?

En general la mayoría de los jóvenes, actualmente, utiliza la tecnología para mantenerse en contacto con sus amigos, jugar, ver vídeos, pero muy pocos crean sus propios juegos, simulaciones o animaciones. Es como si los jóvenes que participan plenamente de la cultura digital “pudieran leer, pero no escribir” (Resnick, et al. 2009). Son consumidores de la cultura digital, pero no son creadores ni autores de la misma.

Como dijimos anteriormente Scratch se basa en las ideas constructivistas de Logo. El constructivismo implica que el alumno experimente con los objetos o con herramientas que le permitan crear sus propias estrategias para aprender y resolver problemas. El aprendizaje desde el punto de vista del constructivismo es dinámico y activo, de modo que el conocimiento es construido por la persona que aprende. En este sentido, si bien es lícito atender a las diferentes formas y estilos de construir de cada estudiante, no es lícito atribuirle “actividad” al sujeto normal y una “pasividad” receptiva, memorística, refleja al sujeto con NEE (Havlik, 2000, p 21).

Algunos conceptos claves de la teoría constructivista en la que está imbuido Scratch son el “conflicto” y el “sentido del error”. El conflicto es necesario para aprender, si no existiera conflicto no aprenderíamos. Cuando construimos cualquier conocimiento o queremos resolver un problema, pasamos por etapas de desequilibrio y reequilibrio. Para que haya desequilibrio algún hecho tiene que ocasionar un conflicto en el estudiante y ese conflicto se suele producir de forma espontánea. Justamente esto es lo que sucede cuando programamos con Scratch, de repente, algo no funciona como pensábamos. El hecho de que las cosas no funcionen como habíamos planificado origina conflicto y nos hace plantearnos estrategias de solución de problemas. Sin embargo en educación especial, a veces, la posibilidad de que el conflicto ocurra es remota. Algunos chicos con NEE no se preocupan por sus errores, podríamos decir que tienen “bajo nivel de conflicto” (Havlik, 2000).

El sentido del error también es muy importante desde el punto de vista constructivista ya que significa un reto para aprender, no algo que hay que erradicar del proceso enseñanza/aprendizaje. El aprendizaje reproductivo, conductista, busca el éxito y se aprende sobre todo de los aciertos; en el aprendizaje comprensivo y constructivo son los errores los que resultan más informativos. Para el constructivismo el error es valioso de muchas formas, para el maestro porque le revela la forma de pensar del alumno, para el estudiante porque le lleva a volver sobre lo que ha realizado y buscar formas de corregirlo (Havlik, 2000). La interacción con programas flexibles y abiertos, como Scratch, permite poner de manifiesto muchos errores interesantes.

El trabajo con Scratch no se parece al trabajo del aula tradicional. Trabajar con Scratch implica actividad, comunicación e intercambio de ideas, planificación, enfrentarse a errores y plantear estrategias de solución de problemas. Ver Tabla 1.

Tabla 1. Comparación de la situación de aprendizaje planteada al utilizar Scratch y de la planteada en el Aula Tradicional

El trabajo con Scratch	El trabajo en el aula tradicional
El estudiante es activo	El estudiante es pasivo
Comunicación e intercambio de ideas entre estudiantes	Trabajo individual con pocas posibilidades de compartir
El estudiante planifica actividades	El estudiante responde a las actividades planificadas por otros
Cada estudiante trabaja en proyectos de su interés	Los estudiantes trabajan en el proyecto asignado por el profesor
El conflicto y el error son necesarios para aprender	El conflicto y el error tienen un carácter negativo, hay que evitarlos.
Cada estudiante avanza a su propio ritmo	Todos los estudiantes deben conseguir resultados uniformes
El docente no es depositario de todo el saber. Simplemente es guía del proceso de enseñanza/aprendizaje.	El docente es el que sabe y dirige la clase.
El estudiante es cada vez más autónomo	El estudiante es totalmente dependiente

En síntesis, el trabajo con Scratch ofrecerá a los estudiantes con NEE oportunidades de construir activamente sus conocimientos, planificar proyectos, plantear dudas y preguntas y trabajar en la resolución de problemas, todo ello les permitirá un aprendizaje activo y significativo.

Otro concepto importante es que al trabajar con Scratch, los alumnos con NEE se igualarán a otros en el error, en el hecho de que todos nos equivocamos, además aprenderán que los errores tienen sentido, sirven para aprender y se pueden corregir. Havlik (2000), en este sentido comenta: cuando un chico con NEE se da cuenta de todo esto, puede tomarse la libertad de decir: “*me equivoqué, ¿cómo se arregla esto?*”, en lugar de: “*soy tonto, no me sale nada bien*”. Con este cambio de actitud quedan abiertas muchas posibilidades de avance y de progreso. (p. 27)

Cómo hacer Scratch accesible

Los autores de este artículo después de conocer la filosofía de Scratch y familiarizarnos con su uso, fuimos conscientes de las múltiples oportunidades que este lenguaje de programación ofrece para los niños y jóvenes con diferentes habilidades y estilos de aprendizaje, pero también comprobamos que el interfaz de Scratch no estaba diseñado de acuerdo a los principios del “Diseño Universal para el Aprendizaje” (en adelante DUA) (Sánchez-Montoya y López-Escribano, 2008).

Un software cumple los principios del DUA si es accesible a todos los posibles usuarios (Maguire, J.M., Scott, S.A. y Shaw, S.F., 2006)

¿Qué ocurre cuando un software no es accesible? ¿Qué podemos hacer cuando un alumno, por ejemplo, desea utilizar Scratch y no puede utilizar el teclado aunque sí tiene movimientos voluntarios para controlar un joystick o puede emitir mensajes orales a un micrófono? ¿Cómo conseguimos que utilice el teclado y que participe en la actividad como el resto de sus compañeros? ¿Desistimos o hay alguna alternativa?

Algunas respuestas a estas cuestiones las encontramos en las Rampas Digitales o Assistive Technology que, aunque no son la solución ideal, sí suponen un camino de esperanza, cuando los medios estándares –teclado, ratón, pantalla o impresora en tinta– no son accesibles a las necesidades de un determinado alumno. Además en muchos casos estas adaptaciones son gratuitas. En la Tabla 2 ofrecemos un resumen de herramientas que se pueden utilizar para conseguir que Scratch sea accesible a estudiantes con NEE.

Tabla 2. Ejemplos de Rampas Digitales (Sánchez-Montoya, 2011)

Producto	Windows	GNU/ Linux
Adaptar teclado y ratón	Escritorio, Opciones de accesibilidad, <i>Emuclit</i> ¹	Escritorio de Gnome,
Webcam, el ojo que todo lo ve	HeadDev ² , CameraMouse ⁴ , HeadMouse ⁵	FacialMouse ³ ,
Joystick sustituye al ratón	MouseJoystick ⁶	
Reconocimiento de voz	Dragon Dictate ⁷	
Teclado sustituye al ratón	MouseKeys ⁸	
Leer con los oídos	Jaws ⁹ , NVDA ¹⁰	Orca ¹¹

¹ Gratuito : www.antoniosacco.com.ar/soft/emuclit16_inst.exe

² Gratuito : www.integraciondiscapacidades.org/index.php?m=Descargas&op=descargar&did=3

³ Gratuito : www.capacidad.es/FacialMouse-v2.2

⁴ Gratuito : www.cameramouse.org/downloads/CameraMouse2010Installer.exe

⁵ Gratuito : <http://robotica.udl.es/headmouse/headmouse2/download/HeadMouse2Installer.exe>

⁶ Gratuito : www.xtec.cat/~jlagares/download/mousejoystick.zip

⁷ Más información : <http://www.nuance.com/naturallyspeaking/products/default.asp>

⁸ Gratuito: www.microsoft.com/spain/accesibilidad/training/windowsxp/mousekeys.aspx

⁹ Demo: <ftp://ftp.freedomscientific.com/users/hj/private/WebFiles/JAWS/J11.0.756-32bit.exe> .

Producto	Windows	GNU/ Linux
Magnificar la pantalla	Zoomtext ¹²	Orca ¹¹
Leer con los dedos	Línea braille ¹³ , Impresoras braille	
Activar / desactivar uno o varios conmutadores		
a. Teclado virtual en pantalla	Click-N-Type ¹⁴	
b. Barrido intencionado por la pantalla	Kanghoo ¹⁵	Java Kanghoo ¹⁵
c. Selección de programa por línea horizontal/ vertical.	Screen Scanner ¹⁶	
Diseñar Tableros de comunicación	Plaphoons ¹⁷	Javaplaphoons

Experiencias del uso de Scratch con personas con NEE

Seguramente existen numerosas experiencias sobre la utilización de Scratch con alumnos con NEE, pero desafortunadamente la mayoría no han sido documentadas. No obstante, a continuación relataremos algunos ejemplos que nos parecen ilustrativos y que pueden animar a los maestros y educadores que trabajan con estos estudiantes a utilizar Scratch en sus clases.

Discapacidad intelectual

La experiencia que relatamos a continuación tuvo lugar en la “Feria de Realidad Virtual, Ocio Electrónico y Discapacidad” celebrada en Salamanca en el 2007. Allí presentamos Scratch a un grupito pequeño de jóvenes con discapacidad intelectual. Les enseñamos la pantalla inicial en la que aparece el gato de Scratch. Utilizamos los bloques de movimiento (color azul) concretamente el bloque “*mover X pasos*”, desplazamos este bloque al área de programación y de modo muy sencillo, con el teclado, introducimos diferentes valores en la variable “X” de este bloque. Así, conseguimos que el gato se desplace tantos pasos como queramos. Dependiendo de si el número introducido es grande o pequeño, el gato se desplazará más lejos o más cerca.

Una vez explicadas estas sencillas instrucciones, dejamos probar a estos jóvenes, ellos introducían el número y hacían que se desplazara el gato. Enseguida fueron conscientes de que si el número era pequeño el gato se desplazaba poco y si era más grande avanzaba más. Comprobamos que les gustaba experimentar con los números de este modo. De repente, a uno de ellos, que estaba viendo que el gato se estaba aproximando al borde de la pantalla, le surgió una pregunta “*¿qué pasaría si el gato llega al borde de la pantalla, se chocaría o desaparecería?*”, de este modo

¹⁰ Gratuito: http://www.nvda-project.org/download/releases/nvda_2009.1_installer.exe

¹¹ Gratuito: <http://www.tiflolinux.org/>

¹² Demo: <http://www.aisquared.com>

¹³ Más información: <http://cidat.once.es/>

¹⁴ Gratuito: <http://www.lakefolks.org/cnt/Spanish-Files.exe>

¹⁵ Gratuito: <http://www.xtec.cat/~jlagares/download/kanghoo.zip>

¹⁶ Gratuito: www.xtec.cat/~7Ejlagares/download/screenscanner.zip

¹⁷ <http://uploadingit.com/file/t7fmbmsyc3vpi804/plaphoons.zip>

sencillo y sin buscarlo específicamente, este alumno acababa de plantear una hipótesis de trabajo. A continuación decidimos probarla. Al hacer que el gato avanzara más éste desapareció ante el asombro de todos los jóvenes asistentes a esta sesión. Y ahora “¿cómo hacemos para que regrese?”, de este modo, también espontáneo, comenzamos la solución de problemas. Durante la hora que aproximadamente duró esta sesión, surgieron más preguntas y más actividades, el grupito de jóvenes con discapacidad intelectual estaba entusiasmado, sentían que podían controlar lo que estaba pasando y esto les animaba enormemente.

Lo que queremos mostrar con esta experiencia es como Scratch plantea un modo diferente de enseñanza/aprendizaje, con Scratch el alumno es protagonista, necesita pensar, puede plantear preguntas y soluciones a esas preguntas. El aprendizaje es activo y constructivo.

Autismo

En la actualidad, hasta donde nosotros conocemos, hay publicadas o recogidas la dos experiencias que han utilizado Scratch con estudiantes autistas y con Síndrome de Asperger (Adams, 2010; Gold, 2011).

A continuación presentamos la experiencia relatada por Adams (2010) “Bob es un estudiante de 14 años que atiende un campamento de verano. En este campamento una de las actividades es la creación de proyectos utilizando Scratch. Se trata de un joven que generalmente tiene buenas habilidades de comunicación, aunque, a veces, tiene arrebatos verbales tanto cuando está muy contento, como cuando está frustrado, lo que le ocurre frecuentemente cuando está aprendiendo a programar. Raramente participa en las actividades durante los descansos después de las comidas, pero hace una excepción cuando se trata de discutir sobre Scratch, juegos de ordenador o películas. Pronto sus compañeros se dan cuenta que es un poco diferente, pero esto no establece una barrera entre ellos. Los proyectos que Bob realiza con Scratch son ambiciosos juegos interactivos, en los cuales, en algunas partes de la historia, el jugador guía al héroe a superar obstáculos. A Bob le entusiasmó utilizar Scratch, durante el campamento de verano, lo que menos le gustó es que llegó el final y no pudo terminar su proyecto, aunque hay que decir que se quedó más tranquilo cuando le dijeron que podía continuarlo en casa” (p. 359-360)

Gold es una profesional de la educación que colabora con el “Marshfield School” en una clase extraescolar destinada a alumnos con necesidades educativas especiales (espectro autista) de 8 a 12 años. En una entrevista reciente, realizada por el “ScratchEd Team” (2011), relata su experiencia sobre la utilización de Scratch con alumnos con Síndrome de Asperger y autismo: “*mi primera clase con Scratch fue bastante caótica. Antes de darme cuenta, la clase entera estaba revuelta llorando, pegando patadas y gritando. Suerte que la directora de Educación Especial vino a salvarme*”.

Después de aquella clase tan difícil, Gold cambió su estrategia a una metodología más individualizada. Ella y algunos mentores seleccionados, entrenan a los estudiantes uno a uno en la utilización de Scratch. La respuesta está siendo positiva. Gold comenta: “*me he dado cuenta que estos estudiantes no quieren que se les apoye*

tanto. Inicialmente planifiqué cuidadosamente cada clase y comprobé que ninguno de ellos me seguía. Su spam de atención es muy corto y es muy difícil. Actualmente entreno uno a uno, sentándome con cada uno de ellos. Hay también ayudantes. Tengo chicos con Asperger de educación secundaria trabajando conmigo, son muy buenos ayudando a los otros”.

Gold (2011) en uno de los foros de Scratch dice como inmediatamente comprobó *“la irresistible conexión entre estudiantes con espectro autista y la programación. Aunque se piensa que la programación es un acto solitario, puede ser para estos estudiantes la primera experiencia social y puede servir como un plan potencial y viable de carrera para ellos. Los estudiantes con Síndrome Asperger y espectro autista con alto funcionamiento, encuentran una zona de confort e interés que no habían descubierto antes. Estos estudiantes pasan de ser invisibles a ser reconocidos a través de su trabajo que los transforma en superestrellas creadores de juegos. Estos jóvenes descubren un nuevo medio que captura su interés de un modo tan absorbente que fascina verlo.*

Discapacidad motriz

Rocío es una estudiante con parálisis cerebral y unas condiciones neuromusculares que le afectan a sus movimientos y posturas. No puede caminar y sus manos no son del todo funcionales. Es usuaria de silla de ruedas de conducción eléctrica. Algunas personas la ven sentada en su silla de ruedas y piensan que no puede hacer nada. Rocío, sin embargo, no acepta estas expectativas tan bajas y está decidida a buscar formas de superar sus limitaciones. Ella utiliza el “Mousekey” y un teclado numérico independiente para utilizar Scratch (Figura 1). Sus profesores desean aprovechar el carácter interactivo de este programa para motivarla y mejorar sus competencias.



Figura 1. Rocío, está utilizando el MouseKey para interactuar con Scratch.

El proceso de adaptar la tecnología mediante Rampas Digitales es lento y sólo podemos considerarlo acabado cuando observemos y valoremos cómo lo utiliza el alumno y creamos que hemos conseguido lo que deseábamos. Durante todo este tiempo, nuestra complicidad con el ordenador transcurre en una atmósfera entre amigable y hostil. Cuando la tarea se hace tediosa, surge la tentación de abandonarla pero, si continuamos, podemos llegar a realizar las adaptaciones necesarias para mejorar la autonomía del alumno. Es un continuo

ensayo porque siempre hay algo que mejorar (Sánchez-Montoya, 2011).

Discapacidad visual

Adams (2010, p. 360) relata la experiencia de Chris un estudiante de 13 años ciego que atendía el campamento de verano por primera vez. Su padre le explicó que su hijo estaba interesado en la música y que le gustaría crear un video musical. ¿Era esto posible para un chico ciego?

Chris tenía un ordenador con un interfaz para Braille, pero Scratch no tiene características de accesibilidad que soporten este interfaz. De este modo que el padre de Chris se unió al campamento y Adams relata como Chris comunicaba sus ideas de lo que quería realizar a su padre y éste trabajaba para implementarlas en la pantalla. De este modo, juntos crearon una interesante historia sobre un platillo volante, un murciélago vampiro y humanos mutantes. La banda sonora era una fascinante mezcla de gaitas, efectos de ciencia ficción y otros sonidos interesantes. Chris dijo que lo que más le gustó fueron los “*sonidos de la librería de Scratch*”.

En la Tabla 2 hemos propuesto diferentes alternativas para personas con discapacidad visual, sabemos que el Zoomtext, amplificador de pantalla, funciona con Scratch y puede ser utilizado por alumnos con algún resto visual. Sin embargo, no hemos probado si se puede utilizar Scratch con otras alternativas o Rampas Digitales propuestas para la discapacidad visual, como “Orca” o “Jaws”. Sería interesante comprobarlo para que alumnos con discapacidad visual como Chris pudieran ser autónomos en la utilización de Scratch.

Un programa elaborado con Scratch para la mejora de la lectoescritura

Otra forma de utilizar Scratch con alumnos con NEE es elaborar programas que puedan ser útiles para el aprendizaje de estos alumnos. En esta línea presentamos el programa FLU-OR (López-Escribano, 2008).

FLU-OR es un programa desarrollado en *Scratch* (MIT-Media Lab., EE.UU. <http://scratch.mit.edu/>) y diseñado para mejorar la decodificación, ortografía y fluidez lectora. En realidad el programa es muy sencillo se trata de la presentación de una palabra o una sílaba en la pantalla del ordenador durante un tiempo determinado, que previamente ha sido programado. La ventaja de Scratch frente a otros programas es que podemos seleccionar presentar una palabra por un tiempo menor a 1 segundo y reducirlo cada vez más, tanto como estimemos oportuno. Se podría utilizar “Power Point” para realizar este ejercicio de presentación de palabras, pero cada estímulo debería ser presentado como mínimo 1 sg., en SCRATCH, sin embargo, cada estímulo se puede presentar por menos tiempo que un segundo, podemos presentar una palabra durante 500 milisegundos ó 200 milisegundos. Estos tiempos son más aproximados al tiempo real que tarda el cerebro en procesar una palabra escrita, por tanto son los tiempos que se deben utilizar en los programas de rehabilitación para la mejora de la ortografía y la fluidez lectora. Si lo deseamos podríamos acompañar estas palabras con su representación gráfica y su sonido. La versión más simple de FLU-OR es la presentación escrita de la palabra.

El canal visual se puede estimular a través de la *exposición repetida* de la *palabra impresa* hasta conseguir que el niño adquiriera una representación estable de la misma (Barker, Torgesen, & Wagner, 1992). El objetivo principal de este tipo de intervención es presentar la asociación grafema/fonema de diversas formas y estimular áreas del cerebro que posiblemente estén *adormecidas*.

Los objetivos del FLU-OR son: (a) desarrollar la precisión y automaticidad en los componentes léxicos y sub-léxicos que subyacen a la lectura; (b) mejorar la fluidez en la identificación de palabras, ortografía y desarrollo del vocabulario; y (c) transformar la actitud de los niños con dificultades lectoras hacia el lenguaje escrito a través de la realización de actividades lúdicas y participativas.

FLU-OR es un programa muy sencillo y totalmente abierto, cada profesor puede elegir las palabras que quiere trabajar y el tiempo que desea presentar cada una de ellas. Las actividades programadas dependen de la imaginación y creatividad del profesor. Su uso es muy flexible, se puede utilizar tanto en grupo como individualmente y con niños de cualquier edad y nivel educativo.

El uso diario de FLU-OR durante breves periodos de tiempo, no más de 15 minutos por sesión, posibilita ampliar el vocabulario, el léxico ortográfico y mejorar la ortografía y fluidez lectora. Cada palabra seleccionada debe ser practicada cada día de modos diferentes, la práctica debe ser secuenciada y adaptada a la capacidad del alumno, sin olvidar añadir a esta práctica diaria un componente lúdico y participativo, estos últimos requisitos son fundamentales para desarrollar en el niño el gusto por la lectura y evitar la sensación de fracaso.

Algunos de los ejercicios que se pueden realizar con FLU-OR durante sesiones de 15 minutos son los siguientes:

- Primer día: presentación de las palabras a todo el grupo utilizando un cañón y lectura de las mismas en voz alta todos juntos.
- Segundo día: presentación las palabras a todo el grupo utilizando un cañón. Cada niño debe encontrarlas en una lista que previamente le ha facilitado el profesor y rodearlas con un círculo.
- Tercer día: presentación de las palabras a todo el grupo utilizando un cañón. Cada niño debe escribirlas y después intercambiarlas con su compañero sentado al lado, para comprobar si las han escrito igual o hay diferencias entre ellos. A continuación comprobarán con la lista de las palabras escritas correctamente cuáles se han escrito bien y cuáles no.

Se podrían realizar muchas actividades y juegos más: unión de la palabra con su dibujo, representación gráfica de la palabra etc. El objetivo es trabajar una lista de palabras durante un tiempo para que estas se graben en la memoria ortográfica visual del alumno.

Diferentes versiones del proyecto FLU-OR han sido compartidas y se pueden encontrar en la galería "*Reading Projects/Proyectos de Lectura*":

<http://scratch.mit.edu/galleries/view/168463>.

La versión de FLU-OR que aparece en esta galería puede servir de modelo para crear el propio proyecto FLU-OR y adecuarlo a las características de los alumnos a los que va dirigido.

Conclusiones

Los objetivos de este artículo han sido, en primer lugar, presentar Scratch un lenguaje de programación que como indican sus autores tiene un “*suelo bajo*”, “*techo alto*” y “*amplias paredes*” y en segundo lugar, reflexionar sobre la posibilidad de que todos los estudiantes puedan utilizarlo, incluso aquellos con NEE. Para ello hemos mostrado experiencias y presentado una lista de herramientas que posibilita hacer Scratch accesible.

Nuestra experiencia con maestros en formación y en activo es que, en general, son reticentes al aprendizaje de los lenguajes de programación, incluido Scratch. Como indicamos al principio de este artículo, el alto nivel de abstracción y la complejidad de los conceptos que deben ser aprendidos para programar es un impedimento serio para muchas personas. Si bien es verdad que nuevos lenguajes como Scratch han intentado superar estos obstáculos, muchos educadores no se sienten cómodos utilizando lenguajes de programación.

Sin lugar a dudas, merece la pena seguir insistiendo en las ventajas que tiene tanto para los estudiantes con NEE, como para sus profesores, dedicar un tiempo del curriculum a realizar actividades de este tipo.

Por todos los motivos expuestos anteriormente y porque además creemos que trabajar con Scratch es una experiencia lúdica y divertida, hemos decidido escribir este artículo, con el fin de seguir animando a los profesionales de la educación a embarcarse en esta tarea de enseñanza/aprendizaje gratificante y realmente beneficiosa para todos los alumnos, incluyendo por supuesto a los estudiantes con NEE.

Scratch es una herramienta que por su idiosincracia posibilita el aprendizaje activo y constructivo, de hecho, es difícil imaginarse una situación de aprendizaje reproductivo o memorístico utilizando este programa. Desde el principio, sin darnos cuenta, al comenzar a utilizar Scratch con un grupo de estudiantes, estamos ya inmersos en una situación de aprendizaje que nada tiene que ver con la tradicional. Para que esto sea posible el profesor debe perder el miedo y la inseguridad y aceptar que él no representa la única fuente de saber en el aula. En la sociedad actual, donde es muy fácil obtener información de cualquier tema, este hecho se hace cada vez más tangible. Al utilizar Scratch, el rol del profesor, varía totalmente, aunque debe mediar y hacer de guía en el proceso de aprendizaje, el profesor pasa a ser un aprendiz más.

Las principales conclusiones de utilizar Scratch con alumnos con NEE es que estos estudiantes se sienten protagonistas de su proceso de aprendizaje, esto les motiva enormemente y favorece su autoconcepto. Es increíble como estudiantes que pensábamos que no podían hacer nada o muy poco, son capaces de planificar, establecer hipótesis y de realizar preguntas que no surgirían en cualquier otro contexto de aprendizaje. Para muchos de ellos, además, el trabajo con Scratch, o programas similares, puede incluso significar una opción de carrera futura.

Scratch es una lengua de programación con un “*suelo bajo*”, es decir, que se puede comenzar con actividades muy sencillas, no es necesario ser un programador

profesional para trabajar y disfrutar con este programa. Además es un programa visual y no hay que escribir códigos, ni largas líneas de programación. Esperamos que todos los motivos aquí expuestos puedan animar a que cada vez a más educadores presenten este tipo de experiencias a sus estudiantes.

Fin de redacción del artículo: 19 de julio de 2012

López-Escribano, C. y Sánchez-Montoya, R. (2012). Scratch y necesidades educativas especiales: Programación para todos. *RED, Revista de Educación a Distancia. Número 34*. Consultado el [dd/mm/aaaa] en <http://www.um.es/ead/red/34>

Referencias

- Adams, J.C. (2010). Scratching middle schoolers' creative inch. *SIGCSE 2010. The 41st ACM Technical Symposium on Computer Science Education* Milwaukee, WI, USA: ACM Press. Consultado el [11/06/2012] en <http://dl.acm.org/citation.cfm?id=1734385>
- Barker, T. A., Torgesen, J. K., & Wagner, R. K. (1992). The role of orthographic processing skills on five different reading tasks. *Reading Research Quarterly*, 27, 334-345.
- Gold, V. (2011). Students with disabilities, supporting literacy. *SCRATCHED Discussions*. Consultado el [22/05/2012] en <http://scratched.media.mit.edu/discussions/researching-scratch/students-disabilities-supporting-literacy>
- Havlik, J.M. (2000). La computadora en la discapacidad intelectual. En J.M. Havlik (coord.) *Informática y Discapacidad. Fundamentos y Aplicaciones*. Ediciones Novedades Educativas: Buenos Aires.
- López-Escribano, C. (2008). Programa FLU-OR. Listas de palabras por ordenador para la mejora de la fluidez lectora y la ortografía. En M.D. Hurtado y F.J. Soto (Coords). *La igualdad de oportunidades en el mundo digital*. Cartagena: Universidad Politécnica de Cartagena. Consultado el [11/06/2012] en http://virtual.tecnoneet.org/documentos/2008/2/2_17.pdf
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., y Eastmong, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 1-15. Consultado el [11/06/2012] en <http://llk.media.mit.edu/papers.php>
- McGuire, J.M., Scott, S.S., & Shaw, A.F. (2006). Universal design and its applications in educational environments. *Remedial and Special Education*, 27 (3), 166-175.
- Radenski, A. (2006). "Python First": A lab-based digital introduction to computer Science. *ITiCSE '06 11th Annual Conference on Innovation and Technology in Computer Science Education* Bologna, Italy -June 26 - 28, 2006

ACM New York, NY, USA. Consultado el [11/06/2012] en <http://dl.acm.org/citation.cfm?id=1140124&picked=prox>

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Siver, J., Silverman, B., y Kafay, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52 (1), 60-67. DOI:10.1145/1592761.1592779. Consultado el [11/06/2012] en <http://llk.media.mit.edu/papers.php>

Sánchez-Montoya, R. (2011). ¿Más avance tecnológico implica mayor inclusión? *VII Jornadas de Cooperación Educativa con Iberoamérica sobre Educación Especial e Inclusión Educativa*. Octubre, 2011, Montevideo, Uruguay.

Sánchez-Montoya, R. y López-Escribano, C. (2008). Disabilities and Scratch. *Scratch @ MIT*. Cambridge, Massachussets. Consultado el [18/06/2012] en <http://events.scratch.mit.edu/conference/2008/program.html>

ScratchEd Team (2011). CSEd Week Feature – Scratch in Special Education: An interview with Vicki Gold of Mashfield High School. *SCRATCHED Stories*. Consultado el [11/06/2012] en <http://scratched.media.mit.edu/stories/scratch-special-education-interview-vicki-gold-marshfield-high-school>