

Diseñando ScratchJr: Apoyo para el aprendizaje en edad temprana

2015-05-01

Diseñando ScratchJr: Apoyo para el aprendizaje en edad temprana

ScratchJr es un lenguaje gráfico de programación basado en Scratch y rediseñado para atender las necesidades específicas de desarrollo y de aprendizaje de los niños entre los grados de kindergarten y segundo de primaria. Este documento describe los objetivos y retos de crear una herramienta que responda a las necesidades de aprendizaje y desarrollo de infantes entre 5 y 7 años, mostrando la trayectoria que condujo al diseño actual de ScratchJr a partir de principios guía y de hallazgos en estudios al usar la herramienta en edad temprana.

ScratchJr es un lenguaje gráfico de programación basado en Scratch y rediseñado para atender las necesidades específicas de desarrollo y de aprendizaje de los niños entre los grados de kindergarten y segundo de primaria. Este documento describe los objetivos y retos de crear una herramienta que responda a las necesidades de aprendizaje y desarrollo de infantes entre 5 y 7 años, mostrando la trayectoria que condujo al diseño actual de ScratchJr a partir de principios guía y de hallazgos en estudios al usar la herramienta en edad temprana.

DISEÑANDO SCRATCH Jr

**APOYO PARA EL APRENDIZAJE EN EDAD TEMPRANA

MEDIANTE LA PROGRAMACIÓN DE COMPUTADORES**

|

Louise P. Flannery

Tufts University

105 College Ave.

Medford, MA 02155

louise.flannery@tufts.edu

|

Elizabeth R. Kazakoff

Tufts University

105 College Ave.

Medford, MA 02155

elizabeth.kazako@tufts.edu

|

Paula Bontá

Playful Invention Company

Montreal, Canada

bonta@media.mit.edu

|

|

****Brian Silverman****

Playful Invention Company

Montreal, Canada

bss@media.mit.edu

|

Marina Umaschi Bers

Tufts University

105 College Ave.

Medford, MA 02155

marina.bers@tufts.edu

|

Mitchel Resnick

MIT Media Lab

75 Amherst St.

Cambridge, MA 02139

mres@media.mit.edu

|

RESUMEN

[ScratchJr](#) es un lenguaje gráfico de programación basado en Scratch y rediseñado para atender las necesidades específicas de desarrollo y de aprendizaje de los niños entre los grados de preescolar (kindergarten) y segundo de primaria. La creación de ScratchJr viene a llenar el vacío existente en tecnologías poderosas para la creación digital y la programación de computadores de la educación en edades tempranas. ScratchJr ofrecerá software tanto para que los infantes creen historias interactivas y animadas, como para que los educadores apoyen su adopción mediante su currículo y recursos en línea. Este documento académico describe los objetivos y retos de crear una herramienta adecuada de programación que responda a las necesidades de

desarrollo para niños entre los 5 y los 7 años; especifica además el rumbo que parte de los principios quía y de los estudios con niños en edad temprana, hasta los diseños actuales de ScratchJr y los planes de trabajo futuro.

EL DISEÑO DE SCRATCHJR

Diseño apropiado para generar desarrollo

El software de programación [ScratchJr](#), actualmente en desarrollo por sus autores [1], está basado en Scratch con el propósito de que lo usen los niños desde el preescolar (kindergarten) hasta el grado segundo (edades 5 a 7 años). Este proyecto busca mostrar cómo, ofreciendo un entorno diseñado para sus necesidades de aprendizaje y desarrollo, los niños en edad temprana pueden cosechar muchos beneficios de la programación. Los siguientes principios guiaron el diseño de muchas características y recursos de ScratchJr como herramienta para que los infantes aprendan de diferentes maneras, creando programáticamente historias animadas e interactivas.

La implementación de cada principio se basa en trasladar, al diseño de características del software que apoyen el aprendizaje y la programación, las expectativas apropiadas, generadoras de desarrollo, para el nivel de rasgos físicos, cognitivos, lingüísticos y socio-emocionales de los niños de los primeros grados escolares.

Piso bajo y (apropiado) techo alto, esto significa que sea fácil iniciarse en la programación con ScratchJr, proporcionando espacios para crecer con los conceptos que van incrementando su complejidad, al tiempo que la herramienta sigue siendo manejable para los usuarios más pequeños.

Paredes amplias, esto es, que permite muchos caminos y estilos de exploración, creación y aprendizaje.

“Cachareabilidad”, esto es, tener la flexibilidad suficiente para permitir la construcción incremental de creaciones y conocimientos mediante la experimentación con ideas y características nuevas.

Jovialidad, que ofrezca una interfaz amistosa, alegre, invitadora y juguetona, expresando un espíritu positivo de exploración y aprendizaje.

Visión general de ScratchJr

El software [ScratchJr](#) comprende una biblioteca de proyectos de usuarios, un editor principal de proyectos (Figura 1) y herramientas para seleccionar y dibujar personajes y escenarios gráficos. En el centro del editor de proyectos, se encuentra la página de la historia, la escena que en el momento se está construyendo. Nuevos personajes se pueden adicionar haciendo clic en el botón más (+) ubicado a la izquierda; el texto se agrega pulsando el botón redondo “ABC” y los nuevos escenarios, pulsando primero el botón más (+) de la derecha y luego el botón cuadrado, que tiene una cadena de montañas. Hasta cuatro páginas, cuyas imágenes en miniatura aparecen en el lado derecho, pueden generarse y jugarse en secuencia como historias con múltiples escenas. El conjunto de personajes incluido en cada página se controla desde el listado ubicado a la izquierda de la página de la historia.

Figura 1: Editor principal de proyectos en ScratchJr

La paleta azul de instrucciones de programación se sitúa en el centro del editor. Los niños despliegan, una por una las categorías de instrucciones, primero haciendo clic en los selectores que se encuentran a la izquierda (A) y luego arrastrando los bloques de instrucciones, desde la paleta azul (B), hasta el área de programación que se encuentra más abajo (C). Encajando allí los bloques se crean programas que se leen y juegan de izquierda a derecha (C).

La rejilla que aparece sobre la página ofrece una unidad de medida concreta y medible para las acciones de los personajes. Puede activarse y desactivarse para ayudar a cumplir un propósito determinado; ej: programar versus presentar un proyecto. Los botones de Bandera Verde (Comenzar) y roja (Detener), inician y detienen respectivamente, la animación programada. Tanto la página como el área de trabajo (script), se muestran como las partes más importantes de la actividad y se enfatizan visualmente mediante tamaño, posición y color [2].

Tanto a los detalles generales de la interfaz como los más finos, se les ha dado forma mediante la observación del trabajo de los niños con Scratch en grupos grandes y pequeños y, con prototipos de ScratchJr.

Hallazgos principales

En unión de los principios de diseño, el trabajo piloto con el software de Scratch existente permitió identificar elementos y diseños esenciales que ScratchJr debía ofrecer o evitar. Por ejemplo, en el estudio piloto, a los niños de segundo grado se les facilitó encontrar los bloques de Scratch con base en el texto de sus etiquetas, lo que no sucedió con los niños de kindergarten o de primer grado. De todas maneras, el alfabetismo continúa siendo una barrera al desarrollo de la competencia con Scratch para la mayoría de los niños entre pre-escolar y segundo grado. Explorar sistemáticamente el efecto de los parámetros numéricos en las instrucciones de movimiento se basó en conceptos que no se introducen hasta grados escolares más avanzados: El sistema Cartesiano coordinado, los números negativos y el tamaño relativo de los números de hasta 500 o más. Habilidades que van desde entender las unidades de medida, pasando por navegar cientos de bloques, hasta llegar a predecir el resultado futuro de una serie de acciones, en el contexto de Scratch, quedaron por fuera del alcance tanto de los preescolares como de los niños de primero y segundo grados.

Algunos conceptos de programación, aquellos con resultados visibles e intuitivos, fueron accesibles para estos infantes. Sin embargo, muchas de las instrucciones de Scratch, ya sea por sí mismas o cuando se ejecutan en agrupaciones comunes, no producían un resultado visible. Esta falta de retroalimentación dificultó el que los niños hicieran, en sus programas, asociaciones entre las instrucciones y las acciones resultantes. Además, el muy bajo nivel de las instrucciones, requiere que el programador descomponga un resultado deseado en varios niveles de sub-componentes, cuando sería posible que los infantes obtuvieran mejores resultados, si las instrucciones y los componentes del proyecto tuvieran un nivel suficientemente alto y requirieran menos niveles de descomposición de su objetivo.

Mientras que las posibilidades ilimitadas de los proyectos de Scratch atrajeron un alto nivel de atención en los niños, algunas funciones, con diseños relativamente acotados, resultaron ser distractores o confundieron a los pequeños, quienes exploran y razonan de manera diferente a los niños mayores. Por ejemplo, los bloques de Scratch permiten parámetros numéricos de docenas de dígitos de largo. Algunos niños se sentían impulsados a digitar parámetros de valores extremadamente largos y que se alejaban de la tarea más rigurosa de determinar un valor resultante de la acción

deseada originalmente. Esta placentera incursión inicial tendía a producir resultados que los menores no podían depurar fácilmente debido a la cantidad de los números involucrados y a la naturaleza de las herramientas para reinicializar el estado del proyecto.

De manera similar muchos niños se desviaban del interés en crear una escena particular favoreciendo el hacer clic en bloques o herramientas que ofrecían retroalimentación instantánea pero que no eran típicamente relevantes para lograr su objetivo. Los niños pequeños que habían llenado sus pantallas con cientos de personajes repitiendo clics en el botón del “Objeto misterioso”, luego descuidaron construir comportamientos para todos estos personajes. A pesar de los diversos retos, los pequeños estaban entusiasmados con la posibilidad de influenciar la apariencia y comportamiento de los personajes en pantalla. El entusiasmo demostrado en los momentos en los que lograron hacerlo muestra el nicho que un programa como Scratch puede llenar o atender, si se diseña apropiadamente para que lo utilicen niños entre los grados de preescolar y segundo de primaria.

Pruebas de los prototipos iniciales de ScratchJr, encontraron que muchos infantes lograban substancialmente mayor progreso con el “cacharreo” significativo, aprendiendo sobre el software y creando mini proyectos, de lo que lograron con Scratch. Por ejemplo, los niños más jóvenes de preescolar, que inicialmente usaron el editor de pinturas de Scratch para llenar sus pantallas con personajes aleatorios, exploraron funciones de programación concretas de ScratchJr (movimiento, cambios de apariencia, sonidos). Los niños mayores, los de grado segundo, aprendieron lo suficiente para hacer escenas animadas sofisticadas en ScratchJr como proyectos de final de año.

Para asegurar lo apropiado del software, para la totalidad de los niños entre preescolar y segundo grado de primaria, algunas características del prototipo requirieron más adaptaciones. Fue claro que las acciones de programación deben nos solamente producir un resultado visible sino tomar el tiempo suficiente para que funcionen apropiadamente. Lo anterior facilitó considerablemente el aprendizaje exploratorio de las acciones con instrucciones de programación y, apoyó la percepción de los niños respecto al rol que le correspondía en el programa, a cada acción.

Durante las sesiones de verano con ScratchJr, un currículo basado en el trabajo individual y grupal con “Tarjetas de ScratchJr”, usadas para presentar conceptos de programación fundamentales y aplicarlos en un proyecto semi-estructurado, dio buenos resultados para ese grupo de niños de preescolar y primero de primaria. Además, se identificaron una serie de estrategias para apoyar un rango mayor de niños en el manejo del proceso de solución de problemas, a medida que trabajaban en cada actividad. Partiendo del esquema de color de la interfaz, hasta el diseño de íconos intuitivos, pasando por el conjunto de instrucciones y su categorización, las dos pruebas piloto ofrecieron una retroalimentación valiosa para futuras revisiones del software y del currículo en preparación para probarlos, con todos sus estudiantes, en aulas de clase diversas.

DISCUSIÓN DEL DISEÑO

Habiendo presentado los objetivos del proyecto, los principios de diseño subyacentes en los prototipos iniciales de ScratchJr y los hallazgos clave, producto de la observación de casi 800 niños trabajando tanto con Scratch como con ScratchJr, esta sección presenta una discusión más detallada de las características de diseño actuales tanto del software como del currículo de ScratchJr. También de cómo éstas atienden los objetivos y retos de crear ScratchJr como un ambiente de programación apropiado para estimular el desarrollo en niños de edades entre 5 y 7 años. A medida que se construía, a partir de muchos recursos o funcionalidades poderosas de la versión existente de Scratch, se hizo un rediseño extenso y diferenciado de ScratchJr para que cumpliera con las necesidades y metas de aprendizaje de las aulas de clase desde preescolar hasta segundo grado de primaria.

ScratchJr busca que durante el proceso de diseño, niños tan jóvenes como los de 5 años, exploren significativamente y ganen independencia, en la creación iterativa de historias animadas e interactivas teniendo en consideración los rasgos cognitivos, físicos, de lenguaje y socio-emocionales, de este rango de edad. La disposición de los elementos en pantalla es eficiente pues tiene una tercera parte menos de bloques de los que tiene Scratch, solo algunas otras herramientas esenciales y están ausentes los complejos menús de navegación. Las funcionalidades están dispuestas de manera intuitiva y buscan eliminar movimientos requeridos del ratón (mouse), clics y

desplazamientos. Los elementos de la interfaz son lo suficientemente grandes, para facilitar apuntarle en un computador a bloques y botones con el puntero del ratón o en una tableta con la yema del dedo. Estrategias comunes para desarrollar el alfabetismo temprano a partir del preescolar hasta el segundo grado están al alcance de los niños cuando navegan por su cuenta ScratchJr. Las etiquetas se basan principalmente en íconos, de manera que quienes se están iniciando en la lectura pueden aprender con rapidez los elementos de la herramienta. Cualquiera de los textos, o fue creado por un niño o puede descubrirlo el usuario, en cuyo caso el texto se asocia con el ícono pertinente.

Todos los elementos en un proyecto de ScratchJr, guían el progreso hacia el alcance del objetivo, por lo que deben ser más útiles que distractores. La totalidad del software, el diseño de los colores, la disposición o distribución y la funcionalidad, conducen la atención hacia funciones de programación. La manipulación de las gráficas de personajes se ha implementado como bloques de programación, en lugar de botones no programables. Además, los bloques de programación deben ubicarse en el área de programas para que sean funcionales; pero una vez allí, se les puede hacer clic para se ejecuten o, adjuntarse a un bloque ejecutor, facilitando tanto el “cacharreo” como la programación. Casi todas las instrucciones producen resultados visibles y con la duración suficiente para que los niños las observen; unas pocas, específicas, controlan el flujo o, las instrucciones de nivel superior, se agrupan para minimizar confusiones en niños que todavía no están listos para explorarlas. Este amplio rango de elecciones de diseño busca ofrecer, tanto soporte físico como cognitivo, para ayudar a que los niños usen ScratchJr con algún nivel de independencia que sea razonable para su manejo en el entorno de un aula de clase.

Piso bajo, techo adecuado

En niños de los grados pre-kínder a segundo de primaria, existe una variedad enorme en habilidades cognitivas y en conocimientos. ScratchJr se acomoda a este rango ofreciendo la adecuada altura de techo y un espacio lo suficientemente flexible para crecer más allá de un punto de entrada. Existe variedad en la complejidad de las instrucciones; desde muy directas y concretas a más complejas o abstractas, dando a los niños contenido apropiado para que lo exploren a medida que dominan las

herramientas básicas. De esta manera, los docentes pueden construir un currículo que atienda las necesidades de la clase. Por ejemplo, la primera actividad estructurada en el currículo actual se enfoca en programar movimientos simples para un único personaje con inicio, movimiento y término de instrucciones. Actividades posteriores pueden incluir personajes adicionales, usos más avanzados de las instrucciones de movimiento, combinaciones de acciones y la coordinación de acciones entre personajes.

Mientras muchos aspectos de ScratchJr pueden ofrecer ese rango de complejidad, algunas herramientas y funcionalidades permiten menos flexibilidad y los preescolares y niños de segundo grado de primaria, pueden requerir diseños que son incompatibles. En estos casos y como punto de partida para hacer pruebas, se escogió una solución a mitad de camino, buscando que ofreciera a los niños mayores espacio para explorar sin dejar de lado a los más jóvenes. Por ejemplo, los niños que cursan los grados de preescolar hasta segundo de primaria, se sienten cómodos con rangos de números muy diferentes. Aunque existen muchas soluciones posibles para este tema, se tomó la decisión de establecer un número límite bastante bajo (menos de 25) para los prototipos iniciales de ScratchJr. Los estudiantes de grado segundo están listos para explorar números más altos, pero el beneficio para niños mayores se compensaría con el reto para los más jóvenes.

Un espacio en el medio para “cacharrear”

Una herramienta con un piso bajo eficaz y techos apropiadamente altos, debe proveer además un espacio en el medio para “cacharrear”. “Cacharrear” hace referencia al número de posibilidades con las que un usuario, pueda aprender, de manera incremental, explorando aspectos de una herramienta que no le son familiares y construir un proyecto iterativamente. Si ScratchJr es suficientemente “cacharrearable”, los niños en las edades objetivo, con diferentes grados de habilidad o preparación, podrán encontrar todas muchas funciones y herramientas enriquecidas para aprender mediante la exploración y la creación. Al mismo tiempo, no se verán obstaculizados por diseños o funcionalidades implementadas para atender las necesidades de usuarios que están en otros niveles. Muchas de las decisiones de diseño, en especial referentes a qué instrucciones incluir, el comportamiento producido por cada bloque y

la forma de categorizar el conjunto de instrucciones, se tomaron para cumplir este balance.

El conjunto de instrucciones, contiene mayormente acciones que los niños pueden comprender conceptualmente con facilidad y cuyas acciones son visibles en el comportamiento de los personajes. Estos rasgos ayudan a los niños a conectar conocimiento que les es familiar con nuevo conocimiento y ofrece retroalimentación crucial mediante la exploración. Como se mencionó en la sección sobre el refinamiento del prototipo (4.1.2) [3], los niños aprenden más fácilmente nuevos bloques de programación, cuando pueden ver la conexión entre el bloque del programa y el comportamiento efímero del personaje. Para ofrecer esa retroalimentación todos los bloques de acción toman un tiempo para ejecutarse y se resaltan en el área de trabajo mientras lo hacen. El número limitado de instrucciones con resultados oscuros o instantáneos minimiza los puntos de confusión apoyando el “cacharreo” y el depurado.

Otros ejemplos del diseño para “cacharrear” es asegurar que los conceptos embebidos están dentro de las posibilidades de comprensión del nivel de edad en el que se utilizan incluyendo: el rango limitado de los valores de los parámetros numéricos, puntos de referencia concretos como la rejilla cartesiana para contar unidades de distancia y permitir múltiples maneras de crear resultados similares mediante la combinación de bloques y/o la manipulación en las instrucciones de los valores de los parámetros.

Una amplia bienvenida

ScratchJr se diseñó para ser invitador y accesible para los infantes con diversidad de estilos de pensamiento y creatividad. El esquema de color genera un ambiente juguetón que a la misma vez es conducente a enfocarse en herramientas de programación y en la página que se ha generado para la creación de historias. Las gráficas son brillantes y originales y, las acciones programables, divertidas y significativas. ScratchJr apoya también múltiples maneras de comprometerse con un proyecto. Los niños atraídos por los temas artísticos pueden contribuir o apoyar un proyecto diseñando los personajes o los fondos. Los interesados en la creación de animaciones pueden explorar sistemáticamente o mediante “cacharreo” el rango de

conceptos de programación, basándose en sus estilos personales de pensar, aprender y crear.

La habilidad para crear hasta cuatro “páginas” independientes y para integrar texto y voz en un proyecto, permite a los niños generar sus propios libros de cuentos o historias. Para facilitarles las cosas los niños pueden usar gráficas embebidas o, para adaptar y conectar proyectos con sus propias vidas e ideas, pueden usar herramientas vectoriales de edición, simples, gráficas y escalables, para volver a colorear imágenes ya existentes o diseñar otras nuevas. ScratchJr permite a los niños trabajar para avanzar partiendo de un modo de creación establecido ya sea artístico, programático o narrativo y, proceder de maneras que se acomoden a sus estilos individuales de crear y aprender.

Apoyo para las aulas de clase

Diseñar ScratchJr para usarse en las aulas de clase, entrañó diferentes consideraciones y restricciones, como son objetivos de aprendizaje específicos y una proporción mayor de niños que de adultos, que contrasta a si se usara de manera informal en el hogar o al término de la jornada escolar. ScratchJr se diseñó para posibilitar tanto a niños como a docentes usarlo en el aula y apoyar tres categorías principales de aprendizaje, más allá de la programación misma: estructuras fundamentales de conocimiento, conocimiento disciplinar específico y solución de problemas.

Viabilidad

El software se diseñó para posibilitar a los maestros su manejo y su uso por parte de los estudiantes de los primeros grados escolares de primaria, con intervención mínima de los adultos. En contraste con otros principios de diseño, este objetivo se alcanzo mayormente asegurando que el resultado final o neto de las decisiones de diseño de la interfaz y de la interacción, facilitaran el uso independiente por el mayor número de niños. Lo anterior se evaluó en estudios de clase en los que se anotaron la frecuencia y el propósito de la confianza de los niños en el apoyo de los adultos. Estas observaciones produjeron varios cambios en el diseño de la interfaz y de las interacciones para el uso de herramientas particulares que maximizaban la habilidad

de los niños para descubrir por si mismos nuevas funciones. Por ejemplo, una ruta bastante compleja para borrar un personaje, se simplificó después de que muchos niños, y con frecuencia, pidieron ayuda para recordar el procedimiento. El uso de las cartas de ScratchJr también apuntó a ayudar a los niños a trabajar cada actividad de manera independiente. En general, diseños simples, intuitivos y el “cacharreo” comprensivo de ScratchJr, posibilita el uso independiente y apoya la viabilidad en el aula de clase.

Apoyo al conocimiento fundamental

Las estructuras fundamentales de conocimiento son aquellas que se aplican de manera transversal en todos los dominios; habilidades como: secuenciación, estimación, predicción, composición y descomposición. Programar en ScratchJr puede ayudar a desarrollar esas habilidades de varias maneras. Por ejemplo, los niños exploran los efectos de la secuenciación a medida que componen cada uno de los pequeños programas (scripts) a partir de instrucciones de programación individuales conectadas a los componentes de la historia o de la acción.

A medida que los niños usan los parámetros de los bloques de programación, desarrollan habilidades para hacer estimados, “¿Cuántos?” o “¿Qué tan lejos?”. A los niños se les estimula para predecir qué pasará cuando ejecuten cada iteración refinada de su programa, para pensar si los cambios que han hecho darán por resultado el objetivo que están buscando. En este caso el software ofrece retroalimentación inmediata sobre lo acertado de sus estimaciones y predicciones. Haciendo explícitos estos resultados mediante actividades curriculares y, alentando su práctica frecuente, se puede ayudar a los niños a desarrollar esas habilidades interdisciplinarias y fundamentales o básicas.

Apoyo al conocimiento disciplinar específico

Además de apoyar las habilidades generales de dominio, un aspecto importante de ScratchJr es integrar oportunidades para que los niños aprendan y practiquen conceptos, de los marcos conceptuales estatales y nacionales, en alfabetismo temprano y matemáticas. ScratchJr facilita el aprendizaje del conocimiento disciplinar específico, de dos maneras. Conceptos esenciales de matemáticas y alfabetismo,

contenidos en los marcos conceptuales de los currículos nacionales y estatales para los grados preescolar a segundo de primaria, están embebidos y se practican mediante el uso básico del software. Los maestros pueden también diseñar actividades de ScratchJr que integren y refuercen conceptos de otras áreas que se están estudiando en el aula.

Varios aspectos del formato del proyecto de programación refuerzan el alfabetismo y el conocimiento de los libros que los niños están aprendiendo en el salón de clase. Componentes del software se discuten usando analogías de libros y videos que les son familiares. Los pequeños programas (scripts) se crean y ejecutan de izquierda a derecha, de la misma manera que el idioma escrito. Para apoyar la estructura narrativa, ScratchJr permite a los niños crear proyectos con múltiples páginas tal como sucede en un libro que tiene un inicio, una trama y un final. Textos que muestran el nombre de cada bloque aparecen para apoyar el reconocimiento de palabras permitiendo que los niños, intuitivamente, pareen íconos con el texto relacionado. Cualquier otro texto que aparece en el proyecto ha sido creado por el niño. Aunque en este documento hay una tendencia a resaltar la construcción o el modo “escritura” de la programación en ScratchJr, los niños también “leen” y descomponen los pequeños programas cuando ellos reconstruyen el programa ejemplo contenido en una Tarjeta de ScratchJr, miran sobre el hombro de un compañero o dedican varias sesiones a un proyecto y lo retoman cada día.

Otros aspectos del diseño se enfocan en conceptos matemáticos comunes para los grados de preescolar a segundo de primaria; ante todo, sentido de los números (magnitud) y medida. Dentro de estas áreas, los niños pueden seguir varias rutas de exploración cuantitativa. Se incluyen diferentes tipos de medidas, por ejemplo, distancia, rotación, tiempo e iteración. Cada una apunta a tener una unidad de medida visible e intuitiva, para facilitar la exploración al interior de este contexto. Por ejemplo, la rejilla removible de 20 por 15 cuadrados anteriormente mencionada, se adicionó a la página para que los niños tuvieran una unidad concreta para contar y estimar a medida que determinaban qué tanto mover un personaje. El diseño de unidades de medida menos concretas que no tienen una estructura similar a la rejilla para representar su significado, están todavía en desarrollo, con soluciones simples implementadas para recoger respuestas iniciales y evaluar la comprensión de los

niños sobre los conceptos de medida.

En el uso de la rejilla varias posibles estrategias para programar movimiento de una distancia determinada facilitan la exploración incrementalmente sofisticada de conceptos de números y de programación. Para mover un personaje tres cuadros en la rejilla, el niño puede secuenciar tres bloques “Mover 1”, usando el valor del parámetro por defecto para ese bloque. Pero también puede usar un bloque “Mover 1” y hacer clic tres veces en el pequeño programa o en el botón de Inicio. Finalmente el niño puede cambiar el parámetro de número, creando un bloque “Mover 3”. Las celdas de la rejilla y los ejes numerados, posibilitan estrategias que varían en complejidad, desde estimar y ajustar pasando por contar y sustraer, para establecer parámetros numéricos. La página de ScratchJr tiene veinte unidades de longitud de manera que se anima a los niños a explorar números cuya magnitud tiene significado. Es así como ScratchJr provee múltiples vías para explorar números y medidas en el contexto de tareas o proyectos fascinantes. El trabajo futuro investigará si las variaciones en el diseño, como etiquetas no numéricas en los ejes de la rejilla, pueden dar mejor seguridad de que las herramientas ofrecidas para estrategias más sofisticadas, como restar, no entorpecen o confunden a los niños que están utilizando estrategias más simples como estimar.

Otra de las prioridades del proyecto, ha sido lograr que ScratchJr se adapte fácilmente a las ideas individuales de maestros y niños. Por lo que se incorporaron varios puntos para que los docentes adicionen su propio contenido y diseño de proyectos. ScratchJr viene con un conjunto básico de personajes y escenarios que contrastan con los cientos que están disponibles para Scratch. Esta decisión previene la dificultad que tienen los niños para navegar gran cantidad de opciones, además de estimularlos a ellos y a sus maestros, a crear nuevas gráficas que pueden relacionarse mejor con temas específicos de la clase. Los niños pueden editar las imágenes incluidas o dibujar las propias en un editor de graficas vectoriales escalables embebido en ScratchJr. En una de las versiones evaluadas, los maestros también pueden, en un momento dado, importar imágenes adicionales y generar plantillas de clases e inicios de proyectos, basados en sus objetivos curriculares. Proveyendo flexibilidad para usar las opciones embebidas y un diseño de proyecto en blanco o, adicionando opciones y ofreciendo un punto de partida diferente, tanto maestros como niños tienen herramientas para hacer

actividades y proyectos relevantes para la clase, además de contenidos y objetivos de aprendizaje personalmente significativos.

Apoyo a la solución de problemas

Incluido también dentro del software y del currículo que lo acompaña, existen elementos que apoyan la solución de problemas: a) identificación de un objetivo; b) formulación de un plan; c) desarrollo de una propuesta inicial para alcanzar el objetivo; d) probar, evaluar y compartir resultados; e) depurar y revisar cada uno de los intentos en base a retroalimentación. Muchas de las características de diseño descritas al hablar del piso bajo y la “cacharreabilidad”, apoyan también la solución de problemas. Esto se logra reduciendo cargas cognitivas innecesarias de bajo nivel, liberando recursos mentales para procesos de alto nivel, tales como descomponer una historia por episodios o capítulos o solucionando problemas en un programa que produce resultados inesperados.

Dado que se tienen menos instrucciones de programación que en Scratch, etiquetas de íconos en lugar de texto y un énfasis en instrucciones concretas con resultados visibles y animados, el reto de aprender a diferenciar los bloques de programación y entender su función, se presenta a los niños de cinco a siete años sin comprometer la naturaleza interesante y significativa de los posibles resultados de la programación y de los proyectos. Además, las instrucciones de ScratchJr se seleccionaron para ofrecer unidades naturales de acción que apoyen la descomposición de una historia en partes programables. Un niño de seis años puede crear rápidamente una historia sobre una rana en su hábitat, con instrucciones como “saltar”, en lugar de tener que generar cada salto con instrucciones de bajo nivel.

Estas decisiones de diseño mantienen el reto en un grado de complejidad adecuado y pueden ayudar a los niños a dedicar los suficientes recursos cognitivos a los múltiples procesos de pensamiento de alto nivel involucrados en imaginar y crear un programa. Las tarjetas, elemento clave en el [currículo de ScratchJr](#), también apuntan a apoyar la solución de problemas ayudando a los niños a trazar un mapa de los componentes necesarios para llegar a una solución.

Currículo complementario

Generado en colaboración con docentes, el [currículo de ScratchJr](#) (pdf) apunta a un rango de conceptos que van desde básicos hasta avanzados. Este currículo proporciona a los maestros un formato de clase que pueden implementar de inmediato y adaptar a medida que estén listos para crear sus propias actividades. En el currículo, las actividades introductorias fundamentales, se enfocan en un conjunto de conceptos que siguen una estructura similar. Primero, el investigador o docente, lidera unas actividades de aprestamiento que incluyen variaciones del juego “[Simon Dice](#)”, presenta y demuestra el tema de la lección y dirige la clase en la solución conjunta del problema. Luego, los niños completan el reto usando como guía las tarjetas de ScratchJr, diseñadas en base a las de Scratch. La actividad ofrece un punto de partida para que los niños exploren un concepto o una característica particular de programación y puedan luego expandir la lección en base a sus propias ideas. Los objetivos de las tarjetas de ScratchJr y de otros modelos curriculares que se han ensayado, son apoyar a los niños en el seguimiento de su propio progreso a medida que avanzan en la realización de la actividad, además de modelar formas de estructurar problemas complejos.

Algunas actividades del currículo estimulan también un enfoque no estructurado hacia el aprendizaje y la creación. Durante la exposición inicial a ScratchJr, los niños exploran libremente la interfaz para ver qué pueden hacer tanto esta como ellos. Después de más o menos cinco lecciones estructuradas y durante dos o tres sesiones, los niños aplican sus nuevas habilidades a un proyecto que se enlaza con otro currículo de clase, por ejemplo, el estudio de un autor. Construyendo sobre los diseños del software y del currículo, los maestros tienen bastante influencia en cómo usar y adaptar ScratchJr como herramienta.

CONCLUSIÓN

Este documento ha presentado el entorno gráfico de programación ScratchJr, incluyendo las características e interacciones diseñadas para que los infantes puedan crear escenas e historias animadas e interactivas. El diseño de este software y de los materiales curriculares que lo acompañan, dan la oportunidad a los niños involucrados en el proyecto de participar y comprometerse en actividades de programación de computadores y de solución de problemas, apropiadas para su edad, a la vez que

construyen, a partir de experiencias tradicionales de la edad temprana, tales como: contar historias, razonamiento numérico y espacial, pensamiento creativo y auto expresión. La liberación del software de [ScratchJr](#), del currículo y de la comunidad en línea, ofrecerá a los niños pequeños una nueva y poderosa herramienta educativa y, simultáneamente, una guía para maestros y padres de cómo implementarla para beneficiar diversas áreas del aprendizaje temprano, desde matemáticas y alfabetismo, hasta estructuras fundamentales de conocimiento interdisciplinario [1].

Los niños de las aulas de los primeros grados de primaria, continuarán creando historias, arte y juegos, con muchos medios creativos diferentes. Teniendo a su alcance una herramienta como ScratchJr, además de crayolas, lápices de colores y papel, pueden narrar historias creativas y trabajar en la solución de problemas y alfabetismo digital; todas estas, clave para participar y tener éxito en un mundo incrementalmente digital.

NOTAS DEL EDITOR:

[1] Los autores se refieren a ScratchJr en tiempo futuro debido a que este documento fue escrito y publicado en el año 2013 y la primera versión ScratchJr fue liberada en Julio de 2014, para la plataforma iPad. Actualmente, ScratchJr también se encuentra disponible para Android.

[2] Para obtener información sobre cómo funciona ScratchJr, recomendamos consultar la “[Guía de Referencia de ScratchJr](#)”

[3] dado que esta traducción solo incluye algunos de los apartados del documento original en inglés, recomendamos descargar y leer el [documento completo](#).

CRÉDITOS:

Traducción de apartes del documento “[Designing ScratchJr: Support for early childhood learning through computer programming](#)” escrito por Flannery, L.P., Kazakoff, E.R., Bontá, P., Silverman, B., Bers, M.U., and Resnick, M. (2013). El documento fue publicado originalmente en “Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13)”. ACM, New York, NY, USA, 1-10 DOI=10.1145/2485760.2485785.

La presente traducción no es obra de IDC y no deberá considerarse traducción oficial de dicha publicación. IDC no responderá por el contenido ni por los posibles errores de la traducción.

*Publicación de este documento en EDUTEKA: Mayo 01 de 2015.

Última modificación de este documento: Mayo 01 de 2015.*