

Aprender a programar, programar para aprender

2013-05-01

Aprender a programar, programar para aprender

El Aprendizaje por Proyectos (ApP), en el que juegan las TIC un papel protagónico, es un medio ideal para que los estudiantes adquieran el conocimiento y desarrollen las habilidades para resolver problemas complejos y realizar tareas difíciles. Este módulo compendia una amplia gama de recursos sobre el tema, entre los que se destacan: El Gestor de proyectos de clase, las WebQuests, evaluaciones, un método para seleccionar recursos educativos digitales, artículos, etc.

El Aprendizaje por Proyectos (ApP), en el que juegan las TIC un papel protagónico, es un medio ideal para que los estudiantes adquieran el conocimiento y desarrollen las habilidades para resolver problemas complejos y realizar tareas difíciles. Este módulo compendia una amplia gama de recursos sobre el tema, entre los que se destacan: El Gestor de proyectos de clase, las WebQuests, evaluaciones, un método para seleccionar recursos educativos digitales, artículos, etc.

APRENDER A PROGRAMAR, PROGRAMAR PARA APRENDER

Por Mitchel Resnick, PhD.



Mitchel Resnick

¿Es importante que todos los niños aprendan a escribir? Después de todo, muy pocos de ellos en su vida adulta serán periodistas, novelistas o escritores profesionales. Cabe entonces la pregunta ¿por qué todo el mundo debe aprender a escribir?

Por supuesto, estas parecen preguntas tontas. Las personas usan la escritura en todas las facetas de su vida; para enviar felicitaciones de cumpleaños a sus amigos, para hacer la lista del mercado, para consignar en sus diarios sentimientos personales. El acto de escribir también compromete a las personas en nuevas formas de pensar. A medida que escriben, aprenden a organizar, refinar y reflexionar sobre sus ideas. Claramente, existen razones poderosas para que todas las personas aprendan a escribir.

Yo veo la programación (programar computadores), como extensión de la lectura. La habilidad para programar permite a las personas “escribir” nuevos tipos de cosas, tales como: historias interactivas, juegos, animaciones y simulaciones. Además, como sucede con la escritura tradicional, hay razones poderosas para que todos aprendan a programar.

El reciente surgimiento del interés por aprender a programar, reflejado en sitios Web como “[codecademy.com](https://www.codecademy.com)” y “code.org”, se enfoca en oportunidades de carrera o de trabajo. Es fácil comprender el porqué: el número de oportunidades de trabajo para programadores y científicos de la computación aumenta rápidamente y en esos campos, la demanda supera ampliamente la oferta.

Pero yo veo razones más profundas y generales para aprender a programar. En el proceso de aprender a programar, las personas aprenden muchas otras cosas. No están simplemente aprendiendo a programar, están programando para aprender; pues además, de comprender ideas matemáticas y computacionales, tales como

variables y condicionales, simultáneamente están aprendiendo estrategias para solucionar problemas, diseñar proyectos y comunicar ideas. Esas habilidades son útiles no solo para los científicos de la computación sino para todas las personas sin distinción de edad, antecedentes, intereses u ocupación.

Hace seis años, el equipo de investigación que lidero en el [Laboratorio de Medios del MIT](#), teniendo como objetivo primordial que la programación fuera accesible y atractiva para todos, lanzó tanto el lenguaje de programación [Scratch](#), como su comunidad en línea. Desde entonces, jóvenes de 8 años en adelante, han compartido más de 3 millones de proyectos en el [sitio Web de Scratch](#), además de los miles que se siguen subiendo a diario. Scratch se usa en muchos contextos: hogares, colegios, bibliotecas, centros comunitarios; para muchas edades: desde los grados de primaria hasta la educación superior; y transversalmente en muchas asignaturas: matemáticas, ciencias de la computación, lenguaje, arte y ciencias sociales.

Nos sorprendemos siempre con la diversidad y creatividad que se evidencia en los proyectos. Si le da un vistazo al [sitio Web de Scratch](#) encontrará historias animadas, excursiones virtuales, simulaciones de ciencias, anuncios de servicios públicos, proyectos de arte multimediales, juegos de etiqueta, editores de pintura y, hasta tutoriales interactivos y boletines de noticias.

Como ejemplo permítanme describir algunos de los proyectos creados por una joven “Scratcher” a quién llamaré SaturnoAzul. Cuando SaturnoAzul comenzó a usar Scratch, uno de sus proyectos fue una tarjeta de Navidad con imágenes caricaturescas de Papá Noel y sus renos. Cada uno de estos tenía un instrumento musical que al presionarse tocaba una porción de la canción “Feliz Navidad”. SaturnoAzul envió a sus amigos un enlace a este proyecto como saludo de Navidad.

A medida que trabajaba en su tarjeta de Navidad, SaturnoAzul se dio cuenta de que lo que más disfrutaba era la creación de personajes animados. Así que desarrolló un proyecto en el que caracterizaba una serie de estos: dinosaurios, dragones, caballos voladores. En la sección Notas del Proyecto, animaba a otros miembros de la comunidad para que utilizaran sus animaciones en sus propios proyectos y ofrecía además, crear por pedido, personajes con determinadas características. En realidad, SaturnoAzul estaba generando un servicio de consultoría. Jamás nos imaginamos que

el sitio Web de Scratch se pudiera usar con ese propósito.

Un miembro de la comunidad quería un chita para su proyecto de Scratch, de manera que SaturnoAzul creó un chita animado, basándose en un video que vio en el sitio de la “National Geographic”. Para otro de los miembros de la comunidad creó un pájaro que aleteaba y luego publicó un tutorial, paso a paso, mostrando cómo había generado la animación.

SaturnoAzul se volvió muy conocida entre la comunidad y comenzó a recibir solicitudes para unirse a grupos colaborativos o “collabs” como se les llama en la comunidad Scratch. En uno de esos “collabs”, ella trabajó con otros cuatro jóvenes, provenientes de tres países diferentes, para producir un elaborado juego de aventura. SaturnoAzul creaba los personajes animados mientras los otros miembros del equipo desarrollaban escenarios para el juego, creaban efectos musicales y de sonido y diseñaban los fondos.

En el proceso de trabajar en estos proyectos, por supuesto que SaturnoAzul aprendió habilidades de programación pero también aprendió muchas cosas más. Aprendió, cómo dividir problemas complejos en partes más simples, cómo refinar iterativamente sus diseños, cómo identificar y depurar problemas, cómo compartir y colaborar con otros, cómo perseverar ante los retos que se presentan.

Encontramos que los miembros activos de la comunidad Scratch comienzan a pensar diferente respecto de ellos mismos. Comienzan a verse como creadores y diseñadores, como personas que pueden hacer cosas con los medios digitales, no solo navegar, chatear y jugar. Mientras muchas personas pueden *leer* medios digitales, los “Scratchers” pueden *escribir* medios digitales.

Los miembros de la comunidad Scratch, también comienzan a ver el mundo de manera diferente. Esto lo manifestó un Scratcher de 11 años en un blog público: “Me encanta Scratch. Es más, permítanme re-frasear lo que dije, Scratch es mi vida. He creado muchos proyectos y ahora tengo lo que se llama ‘Mente de Programador’. Esta se manifiesta cuando pienso en cómo está programado todo; desde las tostadoras, hasta los sistemas para carros eléctricos y muchísimo másssss”

The screenshot shows the Scratch 2.0 interface. The stage is titled "Who Do You Like Best?" and has a score of 126. There are four character sprites on the stage: Tara (blue), Nani (brown), Pica (orange), and Giga (red). The "Objetos" (Sprites) panel shows these four characters. The "Scripts" panel contains a complex script for a "Who Do You Like Best?" poll. The script starts with a "When green flag clicked" event, followed by a "say" block for Tara, a "wait" block, and a "say" block for Nani. It then asks the user to click on a character, and based on the click, it updates the score and the "Who Do You Like Best?" text. The script also includes a "say" block for the user's choice and a "wait" block.

(creado por Mitchel Resnick)

Mitch Resnick: *Enseñemos a los niños a programar* (Video TED)

NOTAS DEL EDITOR:

[1] Recomendamos consultar la [Guía de referencia de Scratch 2.0](#). Sin lugar a dudas, Scratch es el entorno de programación de computadores que ha dado más frutos en educación escolar. El 9 de Mayo se lanzó oficialmente la versión 2.0 de este entorno, que ahora funciona completamente en línea. Le invitamos a conocer las novedades y modificaciones más importantes, en comparación con la versión 1.4, consultando la presente Guía de Referencia. Esta hace un barrido rápido y completo de sus principales funcionalidades y comandos.

CRÉDITOS:

Artículo escrito por el Doctor Mitchel Resnick y publicado el 8 de mayo de 2013 en [edSurge](#). El Dr. Resnick es profesor de investigación del aprendizaje en el Laboratorio de Medios del MIT y líder del grupo de investigación [Lifelong Kindergarten](#) en el [MIT Media Lab](#).

*Publicación de este documento en EDUTEKA: Junio 01 de 2013.

Última modificación de este documento: Junio 01 de 2013.*