

¿Por qué trabajar la programación de computadoras en la escuela? Preguntas, sugerencias y herramientas

Por: Lester López
(llopez@usfq.edu.ec)

Las Tecnologías de la Información y Comunicación (TICs) son un motor de impulso para la educación actual. Desde pequeños y con mucha destreza, los estudiantes utilizan las TICs como medio para aprender y compartir experiencias, haciendo honor al término “nativo digital”; término que describe a los estudiantes que han crecido con la tecnología y, por lo tanto, tienen una habilidad innata en el lenguaje y en el entorno digital (García, Portillo, Romo, & Benito, 2007).

En la casa, la sociedad y por supuesto en la institución educativa, los estudiantes tienen contacto con las TICs y aprenden a utilizarlas para resolver problemas de su entorno. En todos los currículos, ya sea como medio u objeto del proceso de enseñanza-aprendizaje, se trabaja con los estudiantes la edición de diferentes tipos de archivos, se desarrolla la competencia de búsqueda y procesamiento de información principalmente mediante Internet, y tal vez en grados superiores se estudian algunos temas relacionados con páginas web, diseño básico multimedia y bases de programación. En la mayoría de los casos el currículo está orientado al uso de las TICs y no a la creación de las mismas. Lo anterior implica que los estudiantes en su mayoría tienen una alta capacidad de uso de la tecnología, pero no quieren entenderla, solo utilizarla.

En los últimos años ha empezado a tomar fuerza una propuesta: ¿por qué no enseñar a los estudiantes desde pequeños a crear sus propias aplicaciones, sus propios juegos? Esta propuesta implica, entre otros temas, iniciar a los estudiantes desde temprana edad en la programación



de computadoras y en el cambio de mentalidad de pasar de ser consumidores de tecnología a emprendedores de tecnología.

“La programación no es para genios” (CodeOrg, 2013), nos dicen Bill Gates de Microsoft, Mark Zuckerberg de Facebook, Jack Dorsey de Twitter, Drew Houston de Dropbox, entre otros personajes destacados en el mundo de la programación, en un video promocionado por la organización Code.org (2013). La programación está en nuestra vida cotidiana y se la puede aprender y enseñar como cualquier otra materia, a través de un currículo de estrategias dinámicas y motivadoras que incentiven la creatividad y el pensamiento crítico, y sea acorde a la edad y la utilización que le dé el profesor.

La programación tiene como objetivo crear programas que resuelvan problemas a los usuarios. Para realizar esto se desarrollan procesos de análisis, diseño, codificación, depuración y mantenimiento del código fuente de programas computacionales.

Escribir códigos de programación implica tener conocimientos de varias áreas, por lo que programar brinda varios beneficios educativos tales como dominar un lenguaje de programación (ya sea gráfico, textual o mixto), desarrollar el pensamiento lógico, ser creativos en la solución de problemas, aprender del error, aprender de forma práctica y divertida, entre otros. Cuando se aprende un lenguaje de programación, como cuando se aprende un lenguaje no nativo, se prepara al estudiante para tener una

visión más amplia de su vida y su entorno. Además ayuda a aprender otros lenguajes con facilidad.

Hay que tener en cuenta que no se trata de formar programadores, sino de formar mentes. En el artículo, *“Teach children how to write computer programmes”* (Livingstone, 2012), se sugiere asociar la creatividad de los estudiantes a la capacidad para darle expresión viable, convirtiendo así ideas en códigos ejecutables, y hacerlo en la época en que los conceptos se asientan en el cerebro de una manera natural.

Tomando en cuenta lo expresado por Piaget y Papert, citados por Falbel (2003), el conocimiento se construye; entonces, la educación consiste en proveer las oportunidades para que los niños se comprometan en actividades creativas que impulsen este proceso constructivo. Tal como ha dicho Papert: “El mejor aprendizaje no derivará de encontrar mejores formas de instrucción, sino de ofrecer al educando mejores oportunidades para construir”.

La teoría del construccionismo afirma que el aprendizaje es mucho mejor cuando los niños se comprometen en la construcción de un producto significativo, como un programa de computación. Así, el aprendizaje involucra dos tipos de construcción: cuando los niños construyen cosas en el mundo externo, simultáneamente construyen conocimiento al interior de sus mentes. Este nuevo conocimiento entonces les permite construir cosas mucho más sofisticadas en el mundo externo, lo que genera más conocimiento, y así sucesivamente en un ciclo de aprendizaje (Falbel, 2003).

Para todo estudiante, ver el resultado de su trabajo produce una gran satisfacción y no se queda solo en la comprensión de la teoría o repetición de problemas ya resueltos. El desarrollo de una programación que resuelva un problema (de preferencia real), ya sea individual o en equipo, desarrolla la autoestima, y la confianza en uno mismo. La motivación es intrínseca, la creatividad aflora y se aprende naturalmente por prueba y error.

Si a esto se suma la motivación externa que pueda brindar el docente, la institución y/o el Ministerio de Educación nacional a través de la preparación, formación de talentos, desarrollo y participación en concursos nacionales e internacionales de programación, robótica, entre otros, se estaría fomentado entonces un uso creativo y emprendedor de la tecnología.

En conclusión, el objetivo de aprender a programar no es en sí mismo escribir códigos, sino desarrollar el pensamiento creativo, el pensamiento lógico, la motivación, la claridad en la comunicación, el análisis sistemático, la colaboración efectiva, el diseño de modelos y el aprendizaje continuo.

¿Podría incluirse entre los contenidos escolares en Ecuador? Considero que sí. Algunos países e instituciones educativas están impulsando este tipo de iniciativas, como por ejemplo Estonia (Tiger Leap Foundation, 2013), Inglaterra (Code Club, 2013), Estados Unidos, Israel, Finlandia, México, Colombia, entre otros. Personalmente las he desarrollado en Ecuador a pequeña escala pero con buenos resultados. Esta inclusión puede hacerse de varias formas, ya sea dentro del currículo o como apoyo a otras materias, siempre tomando en cuenta el desarrollo cognoscitivo de los estudiantes y la preparación del profesorado.

Por último y no menos importante, la programación es una de las salidas profesionales con mayor demanda en todo el mundo (CodeOrg, 2013), y cada vez con mayor trascendencia e influencia en el futuro. Lo que ahora puede ser una clase divertida o un juego para los estudiantes, en pocos años puede convertirse en su carrera profesional y en el motor de desarrollo de un país.

A continuación se muestran algunas herramientas que pueden ser útiles en la introducción de la programación en los niños y jóvenes. Las he categorizado según algunos criterios que pueden ser de ayuda para los interesados en el tema.



Referencias bibliográficas

- Alice. (2013). *Sitio oficial*. From <http://www.alice.org/index.php>
- Brenky. (oct de 2001). Digital natives, digital immigrants. *On the horizon*, 9 (5), pp. 1-6.
- C-Jump. (2013). *Sitio oficial*. From <http://c-jump.com/>
- Code Club. (2013). *Sitio oficial*. From <http://www.codeclub.org.uk/>
- Codea. (2013). *Sitio oficial*. From <http://twolivesleft.com/Codea/>
- CodeOrg. (2013). *What most schools don't teach*. Retrieved 26 de Marzo de 2013 from http://www.youtube.com/watch?feature=player_embedded&v=nKIu9yen5nc
- CodeOrg. (2013). *Sitio oficial*. From <http://www.code.org>
- CodeOrg. (2013). *Stats. What's wrong with this picture?* Retrieved 26 de Marzo de 2013 from <http://www.code.org/stats>
- Falbel, A. (2003). *Construccionismo*. (M. d. Educativa, Ed.) Costa Rica.
- Hackety Hack. (2013). *Sitio oficial*. From <http://hackety.com/>
- Kids Ruby. (2013). *Sitio oficial*. From <http://kidsruby.com/>
- Kodu. (2013). *Sitio oficial*. From <http://www.kodugamelab.com/>
- LEGO MindStorms. (2013). *Sitio oficial*. From <http://mindstorms.lego.com/en-us/default.aspx>
- Livingstone, I. (2012). Retrieved 26 de Marzo de 2013 from <http://www.guardian.co.uk/commentis-free/2012/jan/11/teach-children-computer-programmes>
- One laptop per child. (2013). *Sitio oficial*. From <http://one.laptop.org/>
- Scratch. (2013). *Sitio Oficial*. From <http://scratch.mit.edu/>
- Squeak - Etoys. (2013). From <http://www.squeakland.org>
- Robomind. (2013). *Sitio oficial*. From <http://www.robomind.net/en/index.html>
- Tiger Leap Foundation. (2013). *Sitio oficial*. From <http://www.tiigrilhy-pe.ee/en/technology-school>

UNIVERSIDAD SAN FRANCISCO DE QUITO



EDUCACIÓN EN LÍNEA*

EN TU TIEMPO
EN TU ESPACIO
EN TU COMPUTADORA
EN TU TELÉFONO
EN TU TABLET

*La Universidad San Francisco te ofrece el sistema más moderno de estudio, una experiencia educativa digital/en línea. Tomas dos módulos de 8 semanas cada semestre y puedes tomar hasta dos materias en un módulo para las carreras de: **Administración Ambiental, Administración de Empresas, Administración de Empresas de Hospitalidad, Comunicación Organizacional y Relaciones Públicas, Educación, Finanzas, Marketing, Sicología y Recursos Humanos.**

VISITANOS: WWW.USFQ.EDU.EC
297 1720
ENLINEA@USFQ.EDU.EC

