

Grandes ideas subyacentes en Scratch

2011-02-18

Grandes ideas subyacentes en Scratch

Traducción del documento Grandes Ideas en el que se exploran tanto conceptos y prácticas computacionales, como el enfoque de aprendizaje basado en diseño. Esperamos que los docentes que utilizan Scratch con sus estudiantes lo encuentren útil y esclarecedor.

Traducción del documento Grandes Ideas en el que se exploran tanto conceptos y prácticas computacionales, como el enfoque de aprendizaje basado en diseño. Esperamos que los docentes que utilizan Scratch con sus estudiantes lo encuentren útil y esclarecedor.

•

GRANDES IDEAS SUBYACENTES EN SCRATCH

Integrantes del <http://ilk.media.mit.edu/index.php> “Lifelong Kindergarten Group”, grupo de investigación del MIT desarrollador de Scratch, viene realizando una serie de “Webinars” [1] para profundizar sobre diversos aspectos de Scratch. El segundo de estos, a cargo de los Doctores Mitchel Resnick y Karen Brennan, estuvo dedicado a plantear <http://scratched.media.mit.edu/resources/four-questions-about-scratch-january-2011-webinar> “cuatro preguntas sobre Scratch”. Con dos de ellas se exploran grandes ideas que subyacen tras este entorno de programación de computadores:

Pensamiento computacional

- *Conceptos computacionales*

- *Prácticas computacionales*
- *Perspectivas computacionales [2]*
- Enfoque de aprendizaje basado explícitamente en diseño

EduTEKA encuentra muy útil y esclarecedor para los docentes que están utilizando Scratch con sus estudiantes el documento <http://scratched.media.mit.edu/sites/default/files/BigIdeas.pdf> “Grandes Ideas”; por tal motivo, lo traduce al español. Adicionalmente, recomienda ver la grabación del Webinar <http://connectpro53415882.adobeconnect.com/p77128601/> “Cuatro preguntas sobre Scratch”.

GRANDES IDEAS

Conceptos y prácticas computacionales y enfoques de aprendizaje



El pensamiento computacional está compuesto por una serie de conceptos y prácticas basadas en ideas del mundo de la computación. Existe un creciente reconocimiento de la importancia del pensamiento computacional para comprender y solucionar problemas en un amplio número de contextos, no solamente en el campo de la ciencia de la computación. La programación puede servir como contexto importante para desarrollar el pensamiento computacional.

En el [taller](http://llk.media.mit.edu/index.php) que ofrece el <http://llk.media.mit.edu/index.php> “Lifelong Kindergarten Group” del [Laboratorio de Medios del MIT](http://llk.media.mit.edu/index.php), se provee un enfoque de aprendizaje basado explícitamente en diseño, para entender conceptos y prácticas del pensamiento computacional, mediante programación de computadores.

**

CONCEPTOS COMPUTACIONALES**

Estos son algunos de los conceptos computacionales a explorar:

Secuencia

Identificar una serie de pasos que son necesarios para realizar una tarea.

Ciclos

Ejecutar la misma secuencia múltiples veces.

Paralelismo

Hacer que las cosas se sucedan al mismo tiempo (dos o más bloques de código que se ejecutan simultáneamente).

****Eventos**

**Una cosa que hace que otra cosa suceda.

Condicionales

Tomar decisiones basadas en condiciones.

Operadores

Soporte para expresiones matemáticas y lógica.

Variables

Guardar, recuperar y actualizar datos.

Listas

Estructuras para organizar una colección de ítems.

PRÁCTICAS COMPUTACIONALES

El proceso de llevar a la acción estos conceptos computacionales se apoya con una serie particular de prácticas computacionales. Estas prácticas de solución de

problemas incluyen:

Incremental/iterativo

Desarrollar un segmento o parte de un programa, ponerlo a prueba (ensayarlo) y luego desarrollar otro segmento de ese mismo.

Probar/depurar

Asegurarse de que las cosas funcionen; y, localizar y arreglar errores.

Reusar/remezclar

Hacer algo en base a lo que otros o usted mismo ha(n) hecho.

Abstracción/modularización

Construir algo grande juntando colecciones de partes pequeñas.

ENFOQUES DE APRENDIZAJE

¿Qué resulta particularmente efectivo para explorar conceptos y prácticas computacionales? Los aprendices deben contar con oportunidades para involucrarse activamente en actividades basadas en:

Diseñar

Crear cosas y no simplemente usarlas o interactuar con ellas.

Interesar

Crear cosas que son interesantes y significativas para el creador.

Colaborar

Trabajar con otros en estas creaciones.

Reflexionar

Revisar y repensar las propias prácticas creativas.

NOTAS DEL EDITOR:

[1] Los seminarios Web (conocidos como Webinars) se utilizan básicamente para realizar eventos en línea y pueden tener uno o varios presentadores y más de 100 participantes. Las plataformas para Webinars facilitan tanto el proceso de invitación, como la interacción entre participantes y la presentación de información a grandes audiencias (principio de comunicación ‘de uno a muchos’).

[2] En el Webinar

<http://scratched.media.mit.edu/sites/default/files/FourQuestionsAboutScratch.pdf> “[Cuatro preguntas sobre Scratch](#)” se presentan tres componentes del pensamiento computacional: conceptos, prácticas y perspectivas computacionales. Los dos primeros se tratan en este documento; el tercero, perspectivas computacionales, está compuesto por “expresar, conectar y comprender”.

CRÉDITOS:

Documento elaborado por [Mitchel Resnick](#) y [Karen Brennan](#); publicado en el portal [ScratchED](#) y traducido al español por Eduteka.



MOTOROLA *Este documento se tradujo con el apoyo de Motorola Foundation, Motorola de Colombia Ltda. y la gestión de la ONG Give to Colombia.*

*Publicación de este documento en EDUTEKA: Marzo 01 de 2011.

Última actualización de este documento: Marzo 01 de 2011.*