

Un artista en cada programador

Por Andreas Koch · 2019-09-26

Además de ser adecuados para introducir la programación en la educación escolar, Code It y p5 también son apropiados para proyectos en muchas otras áreas o temas. Se puede usar en clases de arte para enseñar sobre espacio, color, forma, perspectiva, proporción o estética; en las clases de matemáticas para álgebra y geometría y en las clases de física puede proporcionar simulaciones e ilustraciones.

¿Por qué la programación creativa es un gran comienzo para enseñar informática?*

Por Andreas Koch

En un taller de programación reciente, uno de mis alumnos tenía algo de experiencia con JavaScript. No quería programar con bloques, porque le parecía un paso atrás. Sintió que había superado ese "lenguaje infantil". Para nosotros fue un desafío motivarlo a participar. Más adelante en el taller, presentamos nuestro [editor de Processing](#) y comenzamos a programar animaciones. Al instante se entusiasmó con las animaciones y, de repente, no le importó usar la programación de bloques "infantil".

A menudo nos encontramos con esta percepción de la programación de bloques tipo [Scratch](#) como un juguete para niños más pequeños. Especialmente por parte de niños mayores y aquellos que tienen alguna experiencia con los lenguajes de programación no gráficos. A menudo prefieren usar lenguajes de programación "reales", que son utilizados por adultos y desarrolladores profesionales. Los argumentos sobre las virtudes didácticas de Scratch no funcionan en contra de esa percepción.

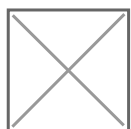
CREAR NUEVAS HERRAMIENTAS DE PROGRAMACIÓN

Con base a estas experiencias, tuvimos la idea de desarrollar herramientas y entornos que combinen las ventajas de la programación mediante bloques con aplicaciones o lenguajes "reales". Aplicaciones que son populares entre los niños (como los juegos), o algo que ellos mismos quieran crear.

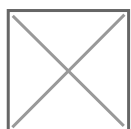
Queríamos crear herramientas de programación que fueran divertidas, fáciles y rápidas de aprender. Herramientas que permiten una expresión creativa abundante, con resultados que deleitan y motivan a los niños. El primer entorno de programación que creamos fue nuestro espacio para crear [Duck Race](#), donde los niños pueden crear sus propios juegos. A la mayoría de los niños les encanta desarrollar juegos, pero las posibilidades creativas de este espacio de creación son limitadas.

En la búsqueda de otras aplicaciones que tienen un gran potencial creativo, descubrimos [Processing](#), que es un entorno de programación diseñado específicamente para artistas. La idea detrás de esto es proporcionar a los no programadores una herramienta con la que puedan crear fácilmente imágenes y animaciones de alta calidad visual y estética. [Processing](#) es, a su manera, un nuevo campo de programación: la programación creativa y la expresión artística a través de la programación.

PROGRAMACIÓN CREATIVA FÁCIL

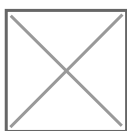


Creamos un editor en el que el procesamiento se puede programar con bloques. Usamos bloques porque facilitan comenzar con la programación. El editor le permite crear imágenes impresionantes con solo unos pocos bloques. Con cada nuevo bloque y cada nuevo concepto de programación, las posibilidades creativas se multiplican. Los niños en nuestros talleres, niñas y niños por igual, estaban entusiasmados con sus imágenes y animaciones. Creemos que esta afinidad con las imágenes se debe a que pueden compartir sus imágenes a través de servicios de redes sociales como Instagram.



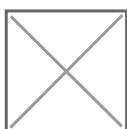
El editor para Processing “[Code it!](#)”, basado en bloques, permite crear [imágenes y animaciones](#) impresionantes

Un programa elaborado en Processing consiste en dibujar formas en un lienzo, produciendo una imagen o una animación. La estructura básica de un programa está compuesta por una función de inicio y una de dibujo. La función Inicio contiene todas las funciones que permanecen estáticas en todo el programa: el tamaño del lienzo, el número de dibujos por segundo, las variables globales, etc. La función Dibujar contiene las formas a dibujar, sus propiedades (color y tamaño, por ejemplo) e instrucciones sobre cómo animarlas. Se dibuja un círculo con solo un bloque. Solo unos bloques más y el círculo ya está animado o sigue la posición del mouse.



Nuestro editor no proporciona todas las funciones de procesamiento.

Siguiendo el método de reducción didáctica, limitamos el editor a las funciones de dibujo esenciales (por ejemplo, formas, bucles, lógica). En el futuro proporcionaremos más funciones que se pueden cargar como extensiones. Se planifican funciones para fotos, videos y sonidos, junto con el machine learning (aprendizaje automático). El editor también permite una transición fácil a la programación de texto. El programa mediante bloques se puede ver como código fuente y se puede exportar al [editor web de Processing](#).



El editor web de Processing “[p5](#)”, basado en texto, permite crear imágenes y animaciones iguales al de la versión de bloques “[Code It!](#)”

En agosto publicamos una [Hora de Código](#) para presentar la programación creativa y nuestro editor de Processing; además de materiales para maestros. Siempre nos complace recibir sugerencias e ideas sobre cómo mejorar la herramienta (info@code-it-studio.de / @_codeit).

INSPIRATE

Además de ser adecuado para introducir la programación en la educación escolar, Processing también es apropiado para proyectos en muchas otras áreas o temas. Se puede usar en **clases de arte** para enseñar sobre espacio, color, forma, perspectiva, proporción o estética; en las **clases de matemáticas** puedes usarlo para álgebra y geometría; En las **clases de física** puede proporcionar simulaciones e ilustraciones. Hay innumerables proyectos en Internet que pueden servir como plantillas: filtros de Instagram, generadores Piet Mondrian, historias en imágenes, animaciones en video, instalaciones de arte o incluso aplicaciones de aprendizaje automático (Machine Learning).

Alrededor de esta propuesta surgió una gran comunidad de artistas, activistas y educadores interesados en Processing. Puede encontrar muchas imágenes, animaciones, proyectos de arte y más en la web, como una gran fuente de inspiración para nuevos proyectos. Los niños pueden formar parte de esa comunidad al publicar sus imágenes y animaciones terminadas en un sitio web, [Instagram](#), [Twitter](#), [Facebook](#) u otras plataformas. Esto les da a los niños retroalimentación y reconocimiento por su trabajo, lo que a su vez los motiva.

[Processing Foundation](#) promueve activamente el uso de Processing con fines educativos: [Saber Khan](#) es el Director de Educación de la Comunidad. Los maestros pueden encontrar una variedad de materiales para la enseñanza e inspiraciones en línea. Por ejemplo, los excelentes libros introductorios y tutoriales de [Daniel Shiffman](#), quien también dirige uno de los [canales de YouTube](#) de programación más entretenidos en la web, o el material didáctico de [CSforALL](#).

UNA GRAN COMUNIDAD DE ARTISTAS, ACTIVISTAS Y EDUCADORES HA EMERGIDO ALREDEDOR DE PROCESSING

CRÉDITOS:

Traducción al español realizada por Juan Carlos López-García, editor del portal Eduteka del artículo "AN ARTIST IN EVERY PROGRAMMER" escrito por ANDREAS KOCH y publicado en el número 10 de la revista Hello World, una publicación de Raspberry Pi (Trading) Ltd., 30 Station Road, Cambridge, CB1 2JH. Andreas Koch enseña CS en escuelas y talleres después de la escuela, y es uno de los fundadores de Code it. El

editor y los colaboradores de Hello World no aceptan ninguna responsabilidad con respecto a cualquier omisión o error relacionado con las habilidades, productos o servicios mencionados en la revista. Excepto donde se indique lo contrario, el contenido de la revista Hello World se publica bajo licencia Creative Commons atribución-No Comercial-Compartir Igual 3.0 (CC BY-NC-SA 3.0). La presente traducción no es obra de Hello World y no deberá considerarse traducción oficial de esta publicación. Hello World no responderá por el contenido ni por posibles errores de la traducción.

Publicación de este documento en EDUTEKA: Septiembre 26 de 2019.

Última actualización de este documento: Septiembre 26 de 2019.