



LENGUAJE DE PROGRAMACIÓN SCRATCH COMO HERRAMIENTA DIDÁCTICA PARA LA ENSEÑANZA DEL SISTEMA DE COORDENADAS CARTESIANAS EN EDUCACIÓN PRIMARIA

Eduardo Gregorio Quevedo Gutiérrez
Víctor Manuel Hernández Suárez
Universidad de Las Palmas de Gran Canaria (ULPGC)
Jacinto Rafael Quevedo Sarmiento
Consejería de Educación y Universidades, Gobierno de Canarias
Alberto Zapatera Llinares
Universidad Cardenal Herrera-CEU, CEU Universities

Resumen

De acuerdo con el currículo educativo actual, la enseñanza del sistema de coordenadas cartesianas en sexto curso de Primaria, constituye un reto para el docente, cuyo objetivo consiste en que los alumnos sean capaces de describir posiciones y movimientos por medio de coordenadas, distancias entre puntos situados en rectas horizontales, paralelismos, perpendicularidad, ángulos, giros, etc., utilizando el vocabulario geométrico [1]. El presente artículo presenta una alternativa didáctica a la presentación del sistema de coordenadas cartesianas haciendo uso del lenguaje de programación visual *Scratch*. La finalidad es doble, por una parte, presentar de forma didáctica el tema tratado y, por otra, introducir a los alumnos en la programación con *Scratch*.

Abstract

The teaching of the cartesian coordinate system in 6th grade, in accordance with the current educational curriculum, is a challenge for the teacher. The objective consists of students being able to describe positions and movements by means of coordinates, distances between points located in straight horizontal lines, parallelisms, perpendicularity, angles, turns, etc., using the geometric vocabulary [1]. This article presents a didactic alternative to the presentation of

the cartesian coordinate system making use of the visual programming language *Scratch*. The purpose is twofold, on the one hand to present the subject treated didactically and on the other hand, to introduce the students to the programming with *Scratch*.

Introducción

El reciente estudio del año 2016 realizado por la Comisión Europea “*Developing Computational Thinking in Compulsory Education, Implications for Policy and Practice*” plantea que, en la última década, el pensamiento computacional y sus conceptos relacionados (por ejemplo, codificación, programación, o pensamiento algorítmico) están recibiendo una atención creciente en el campo educativo. Esto ha dado lugar a una gran cantidad de numerosas iniciativas de implementación públicas y privadas. A pesar de este interés generalizado, la integración exitosa del pensamiento computacional en la educación obligatoria aún enfrenta problemas y desafíos no resueltos [2].

El pensamiento computacional como tal se enuncia a principios de esta década. Jeannette Wing, doctora en Ingeniería en Informática por el MIT, que es una de sus máximas exponentes, presentó una definición en 2011 centrada en el uso de conceptos informáticos para hacer cosas, desde resolver problemas hasta comprender el comportamiento humano, pasando por el diseño de sistemas [3]. El pensamiento computacional se basa fundamentalmente en dos teorías del aprendizaje: el constructivismo de Jean Piaget, psicólogo y pedagogo suizo que defendía la dotación de herramientas al estudiante para que este pueda resolver problemas [4], y el construccionismo de Seymour Papert, matemático, informático y educador estadounidense, nacido en Sudáfrica, que proponía la construcción de modelos mentales para comprender el mundo que nos rodea [5]. Ambas teorías de aprendizaje se centran en la construcción de elementos, siguiendo la filosofía denominada *maker* para resolver problemas.

Scratch es un lenguaje de programación visual desarrollado por un equipo del MIT Media Lab (Instituto Tecnológico de Massachusetts) liderado por Mitch Resnick, que se utiliza por parte de estudiantes, profesores y padres para crear de forma sencilla animaciones e interacciones, que fomentan el pensamiento computacional, poniendo así en práctica las tesis de Piaget y Papert [6]. La principal aportación de *Scratch* es que está destinada a usuarios de temprana edad, lo que la hace directamente aplicable como herramienta didáctica destinada a enseñar programación a alumnos de Primaria. Cada vez más, los alumnos están interesados en la programación como medio creador de diferentes utilidades, aplicaciones y juegos, y este interés aparece cada vez más en edades tempranas. *Scratch* ofrece una perspectiva avanzada en conocimiento, pero sencilla en manejo [7]. Así, actualmente Scratch se está usando en el ámbito educativo para una amplia variedad de aplicaciones [8, 9, 10, 11].

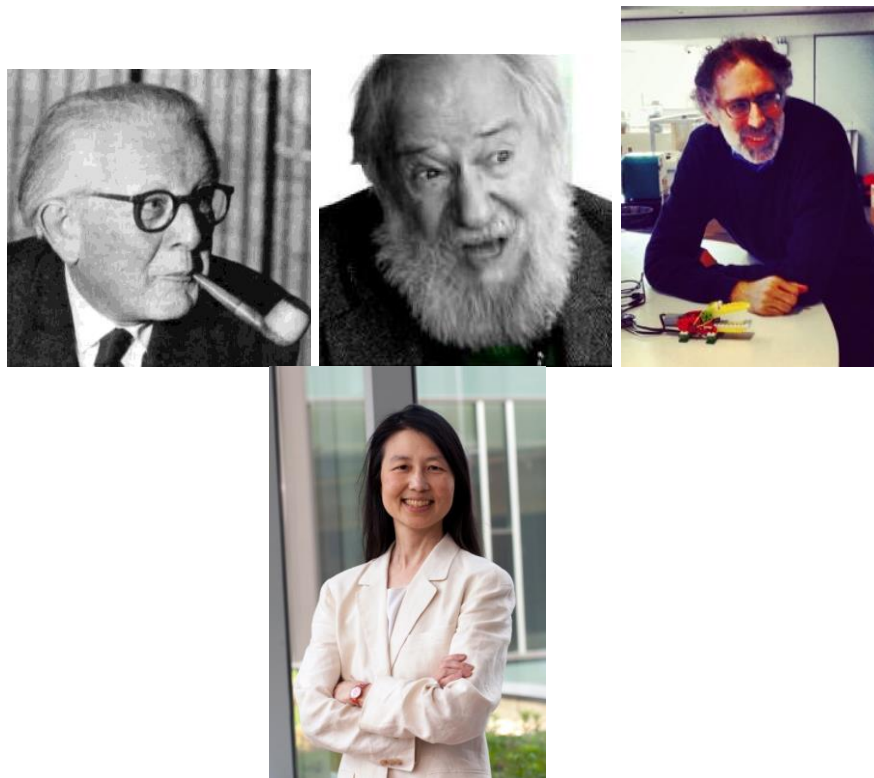


Figura 1. De izquierda a derecha Jean Piaget (1896-1980), Seymour Papert (1928-2016), Mitch Resnick (1956-) y Jeanette Wing (1956-)

El presente artículo presenta una alternativa didáctica a la introducción del sistema de coordenadas cartesianas haciendo uso de *Scratch*. La finalidad es doble: por una parte, presentar de forma didáctica el tema tratado y, por otra, introducir a los alumnos en la programación con *Scratch*.

De acuerdo con el currículo educativo actual, la enseñanza del sistema de coordenadas cartesianas en sexto curso de Primaria constituye un reto para el docente. El objetivo consiste en que los alumnos sean capaces de describir posiciones y movimientos por medio de coordenadas, distancias entre puntos situados en rectas horizontales, paralelismos, perpendicularidad, ángulos, giros, etc., utilizando el vocabulario geométrico.

Enseñanza del sistema de coordenadas cartesianas

La enseñanza del sistema de coordenadas cartesianas en Primaria normalmente se ha orientado a explicar didácticamente la localización en un mapa o gráfico [12]. Se puede trabajar, además, con juegos como, por ejemplo, el clásico “Hundir la Flota” [13]. Tomando en consideración referencias como estas se han propuesto situaciones de aprendizaje como “El Cartesiano” [14].



Figura 2. Fotograma del juego School Age: Sea Battle basado en el clásico “Hundir la Flota” [13]

La alternativa didáctica con *Scratch* propuesta en el presente artículo se centra en comprender el sistema de coordenadas como un elemento propio de la pantalla de ordenador en la que se está trabajando. El escenario de *Scratch*, esto es la ventana de trabajo, se mide en píxeles (px) y su tamaño exacto es de 480 px (ancho) x 360 px (alto). Se puede aprovechar este hecho para explicar a los estudiantes que los píxeles no son más que cuadrados de diferentes colores que componen una imagen digital. Además, se puede hacer ver que una imagen de alta resolución (por ejemplo, de 1920x1200 px) es bastante mayor que el escenario de *Scratch*, tal y como se presenta en la Figura 3 [15].

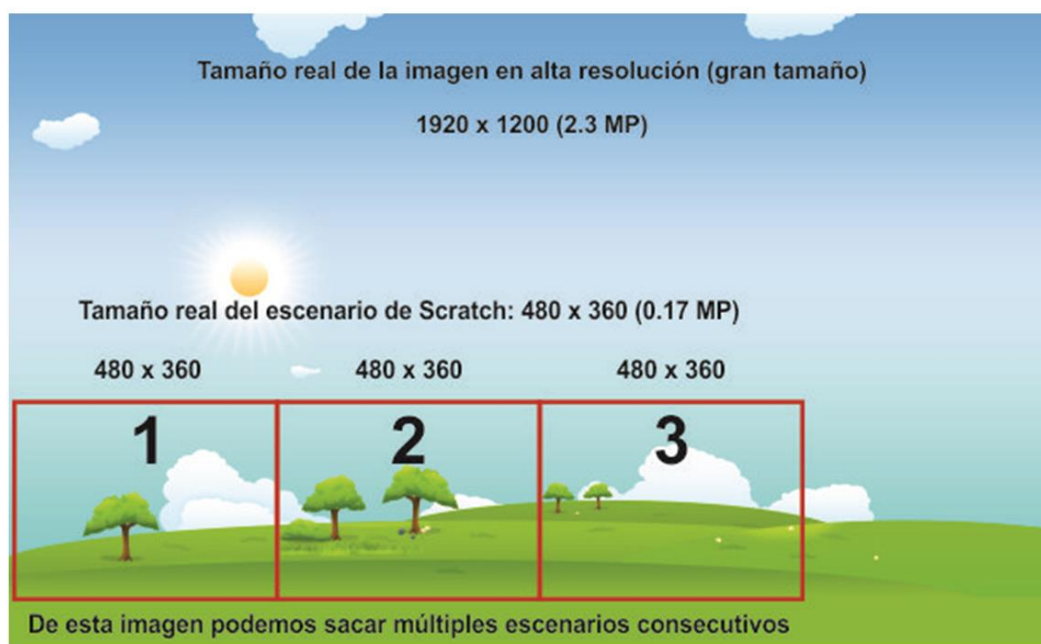


Figura 3. Generación de 3 escenarios de Scratch a partir de una imagen de alta resolución de 1920x1200 px [15]

El sistema de coordenadas cartesianas en *Scratch*

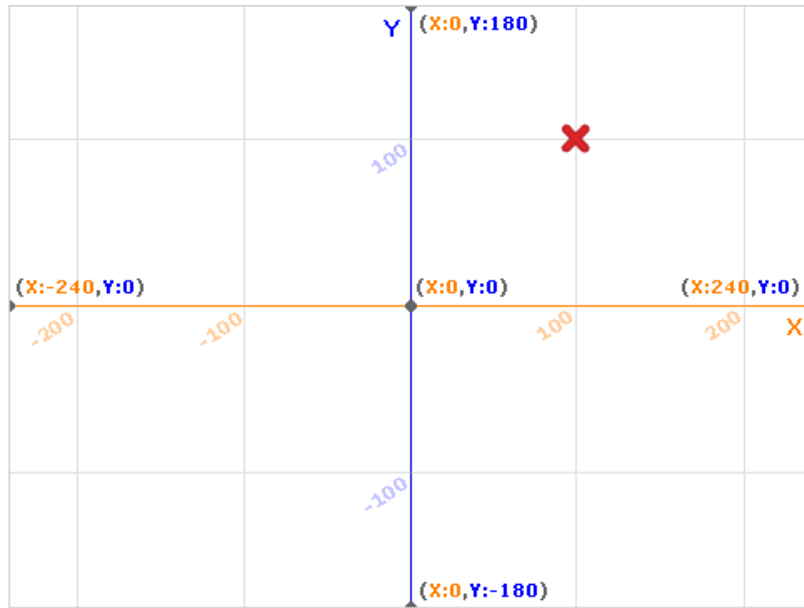
Scratch permite seleccionar una serie de escenarios base como plantilla, o utilizar propios a partir de imágenes seleccionadas por el usuario. Existe una serie de escenarios predefinidos en *Scratch* que presenta el sistema de coordenadas cartesiano de varias formas diferentes:

- Escenario *xy-grid* (Figura 4a): sistema de coordenadas cuyo centro se encuentra en el centro del escenario de *Scratch*. Se presentan los puntos límites en el eje de abscisas: $(-240, 0)$ y $(240,0)$ y en el eje de ordenadas: $(0, 180)$ y $(0, -180)$. Incluye además una cuadrícula de cuadrados de área 100×100 .

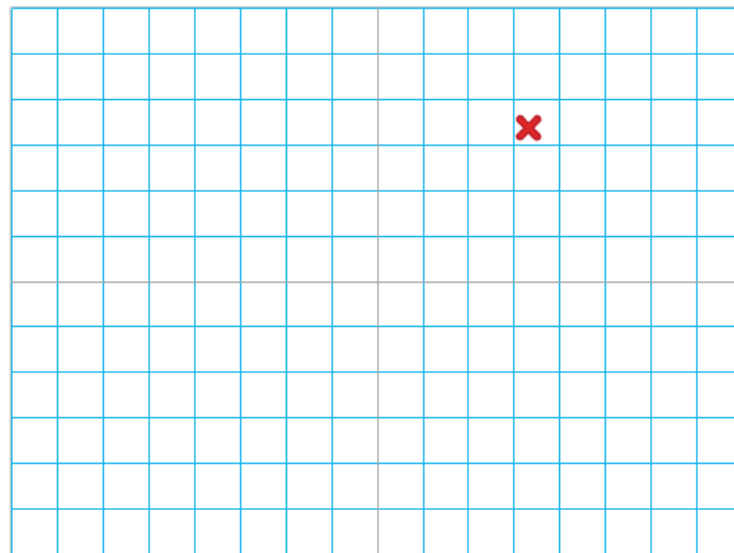
- Escenario *xy-grid-30px* (Figura 4b): cuadrícula de cuadrados de 30×30 . Al ser el escenario de 480×360 px hay 16 cuadrados por fila y 12 cuadrados por columna.

- Escenario *xy-grid-20px* (Figura 4c): cuadrícula de cuadrados de 20×20 . Al ser el escenario de 480×360 px hay 24 cuadrados por fila y 18 cuadrados por columna.

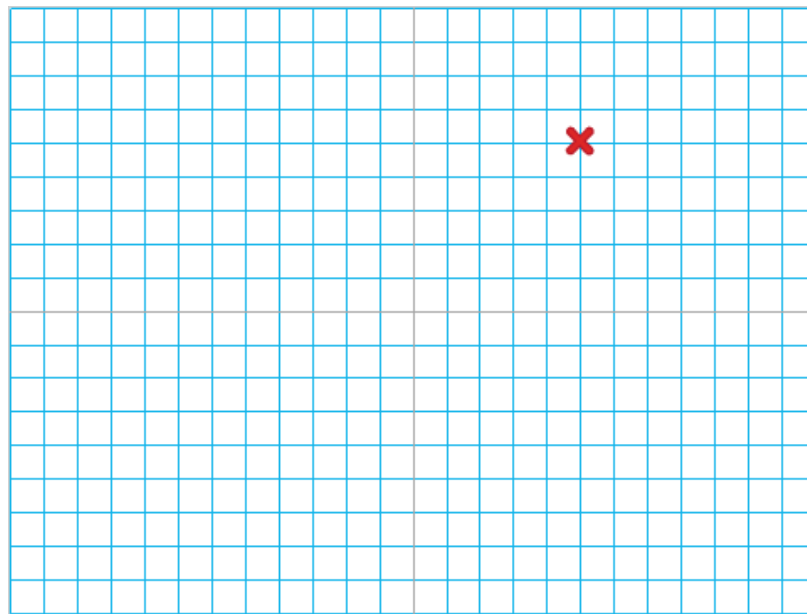
Los escenarios de la Figura 4 son útiles para trabajar el concepto de área de figuras planas, tomando como referencia los cuadrados de 100×100 , 30×30 y 20×20 , respectivamente. Por otra parte, para aproximarnos al concepto de función es adecuado introducir algún objeto a modo de referencia; en la Figura 4 se utiliza el objeto *button 5b*, que presenta una cruz roja localizada en el punto $(100, 100)$. Así, partiendo de un fondo dado, por ejemplo, el de la Figura 4a y un objeto (la cruz roja) se pueden comenzar a trabajar las funciones. Se presentan a continuación algunos ejemplos prácticos:



a) Escenario xy-grid



b) Escenario xy-grid-30px



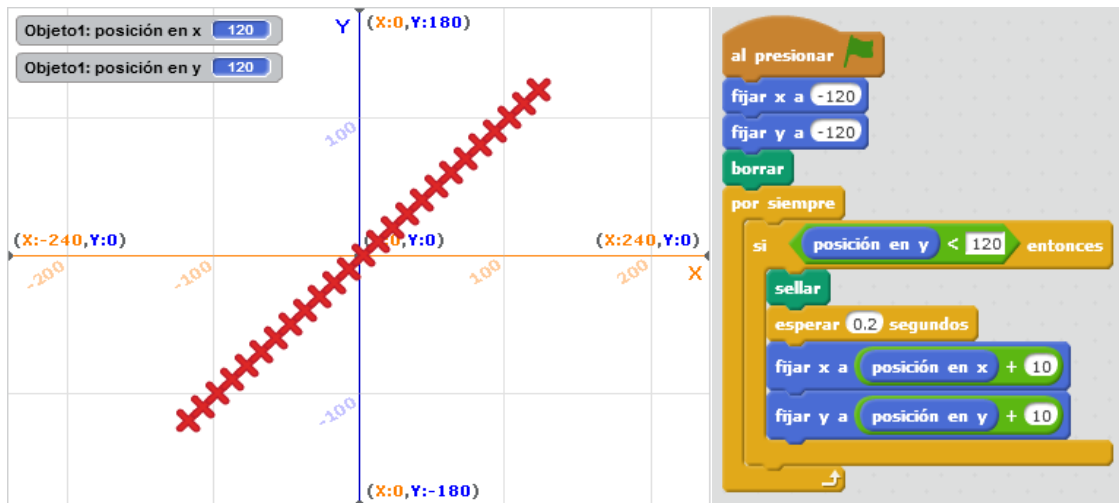
c) Escenario xy-grid-20px

Figura 4. Escenarios de Scratch relacionados con el sistema de coordenadas cartesianas. Como referencia se ha situado una cruz roja (objeto button 5-b) en coordenadas (100,100)

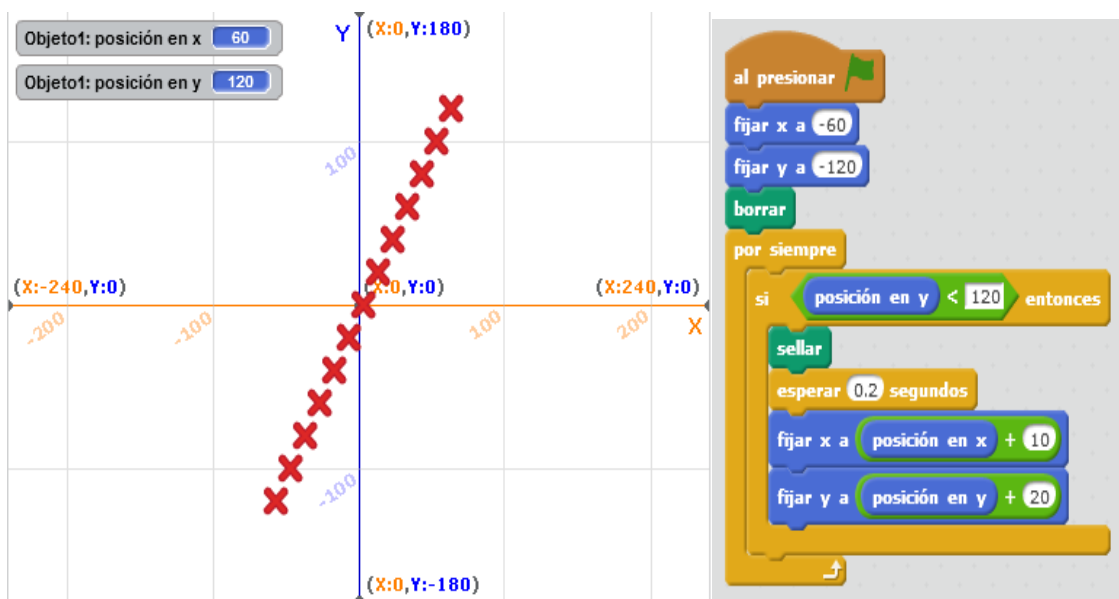
Función lineal

Se presenta un programa en el que inicialmente se introduce una función lineal de pendiente unidad $y = x$. Para favorecer la representación, los saltos van de 10 en 10 tanto para la x como para la y (Figura 5.a). Se trata de considerar que la pendiente no es más que la relación de la función con respecto a los valores de los ejes: $m = y/x$ y verlo de forma práctica. El programa inicializa la x y la y a un valor de -150 y, a partir de ahí, comienza a incrementar ambas variables hasta llegar al valor inmediatamente menor a $y = 150$. Para ello se hace uso de un bucle “por siempre” y de una estructura condicional “si”. Como se puede observar en la Figura 5.a el programa es completamente visual y está organizado en bloques, lo que facilita su comprensión. En el escenario se incluye la posición actual de las variables x e y para favorecer el entendimiento de su evolución.

Si duplicamos o triplicamos el número de unidades que la y crece con respecto a la x , la pendiente en consecuencia también se duplicará (Figura 5.b) o triplicará (Figura 5.c) y el número de cruces dibujadas será menor. Para estos casos se ha de dividir por 2 o por 3 el valor inicial de $x = -60$ para la Figura 5.b y -40 para la Figura 5.c. De esta forma, se generaliza a un caso general de función lineal del tipo $y = m x$. Se observa como el ángulo de la semirrecta generada en el primer cuadrante partiendo de 45° ($y=x$) se aproxima cada vez más al ángulo recto a medida que se incrementa la pendiente (será recto para una pendiente infinita).



a) Pendiente = 1 ($y=x$)



b) Pendiente = 2 ($y=2x$)

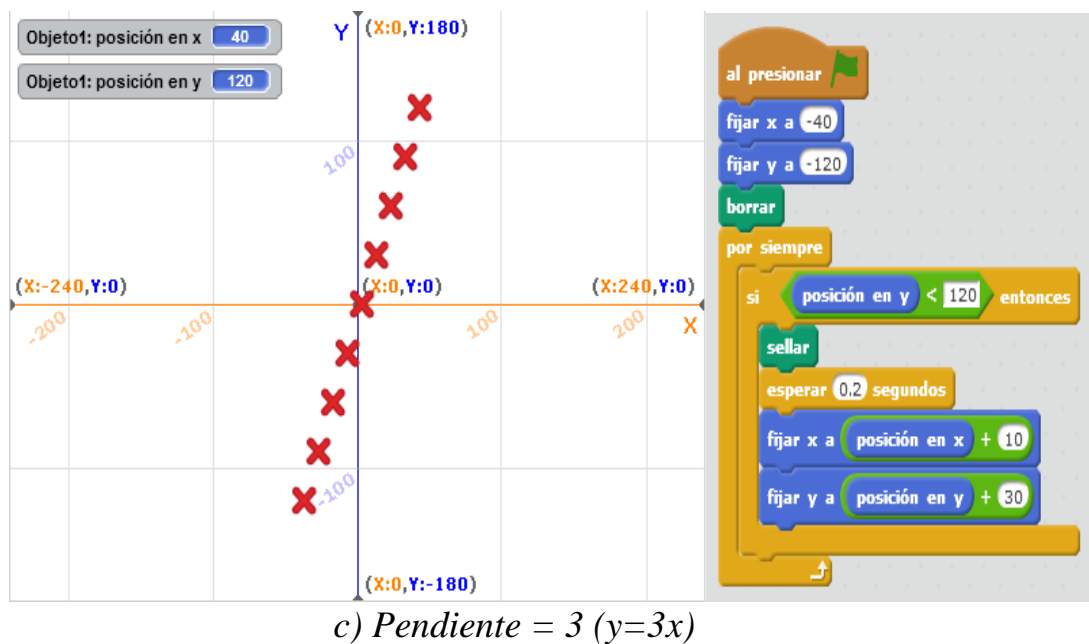
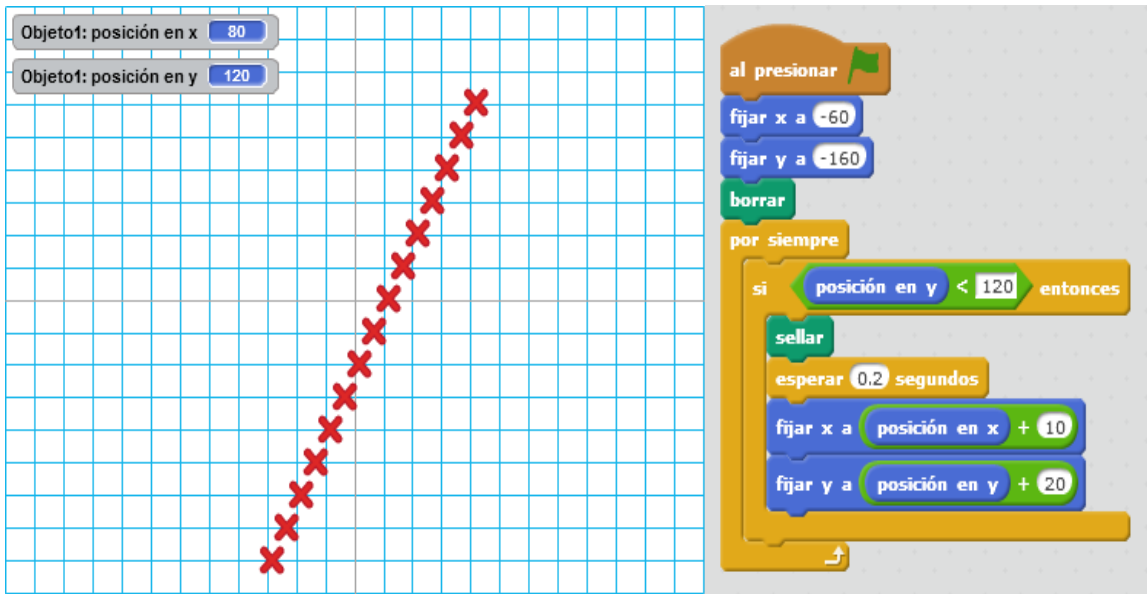


Figura 5. Introducción de función lineal con Scratch

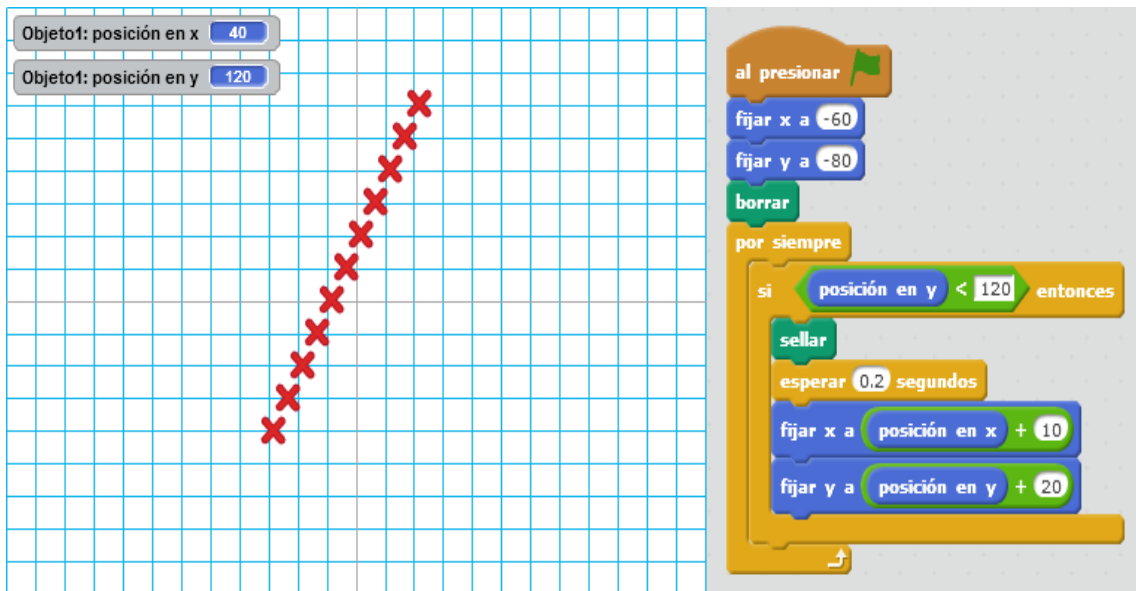
Función afín

Para introducir la función afín $y = mx + n$, simplemente se trabaja sobre uno de los casos anteriores, por ejemplo $y = 2x$ (Figura 5.b) y se comenta que el desplazamiento con respecto a la ordenada (n) es precisamente lo que hace que la función pase de ser lineal a afín, con lo que se genera la idea básica de que la función lineal no es más que un caso particular de la función afín ($n = 0$).

Para representar casos de la función afín, simplemente se desplaza el valor inicial de y , por ejemplo -40 unidades (Figura 6.a) o 40 unidades (Figura 6.b). Esto propiciará que en el paso por el eje de ordenadas se produzca dicho desplazamiento, dando como resultado las funciones $y = 2x - 40$ e $y = 2x + 40$, respectivamente. Para una representación más visual se ha optado en este caso por utilizar el escenario *xy-grid-20px* (Figura 4.c). Se puede aprovechar esta circunstancia para introducir el concepto de paralelismo, ya que, aunque cambia la ordenada en el origen, la pendiente se mantiene constante.



a) $y = 2x - 40$



b) $y = 2x + 40$

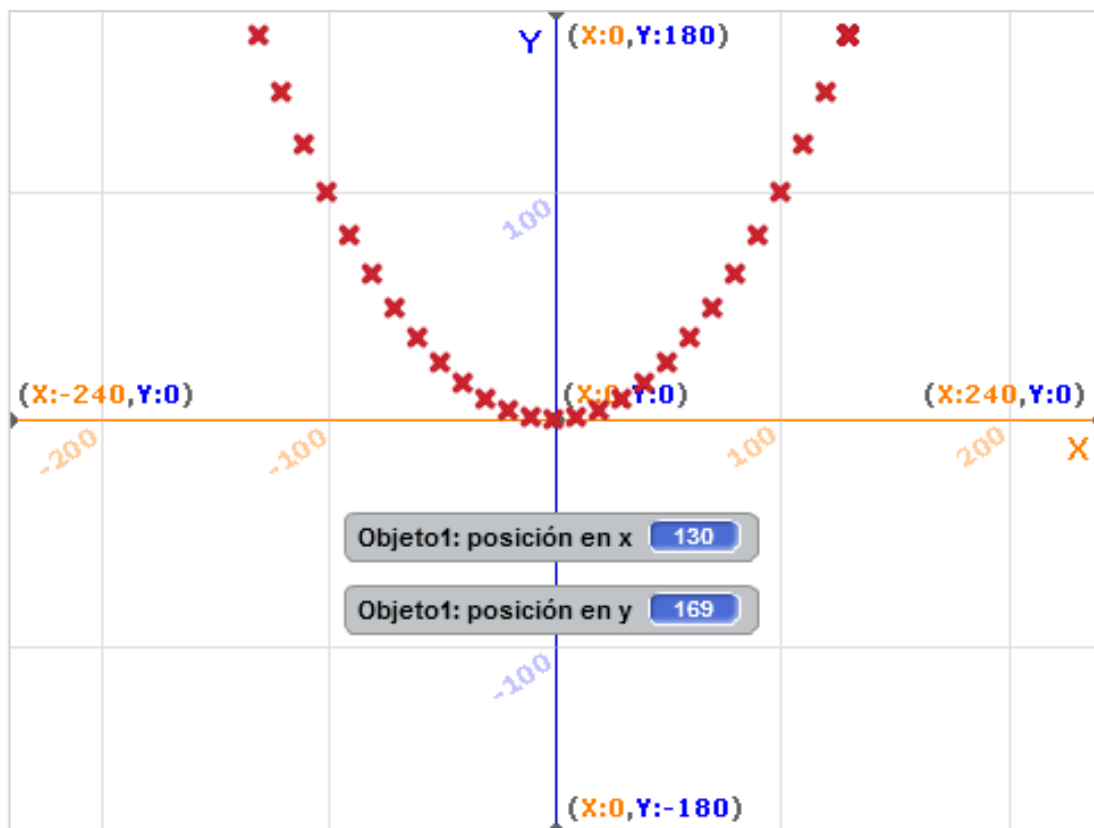
Figura 6. Introducción de función afín con Scratch

Función cuadrática

Para introducir la función cuadrática, en primer lugar, se considera que la representación sea adecuada al escenario de *Scratch*. Para ello se plantean saltos en x de 10 en 10 y un valor de y que compense los saltos de 10 para adaptarlo a la escala del escenario (división por 10 en cada componente de x):

$$y = \left(\frac{x}{10}\right) \cdot \left(\frac{x}{10}\right) = \frac{x^2}{100}$$

De esta forma, esta función se puede reinterpretar simplemente como un ejemplo claro de geometría dinámica [12], en el cual cuando asignamos, por ejemplo, a x el valor de -100 o 100 se pueden visualizar cuadrados de área 100 x 100, tal y como se presenta en la Figura 7.



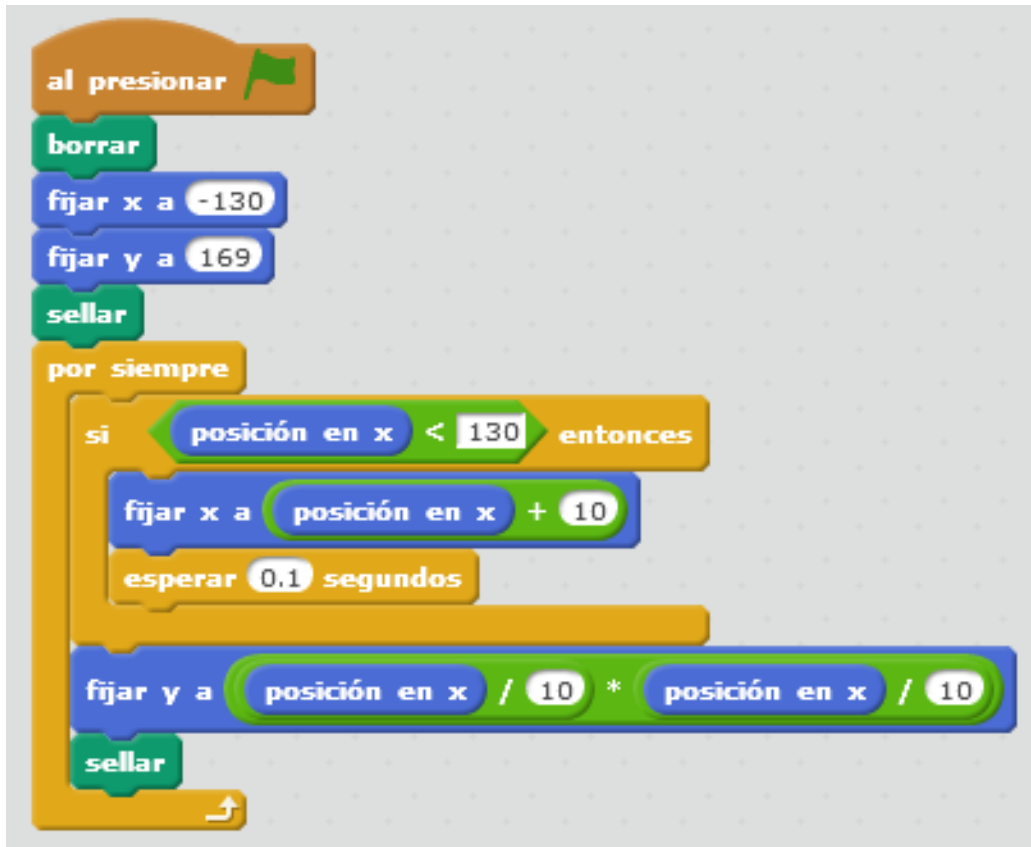


Figura 7. Introducción de función cuadrática con Scratch

Conclusiones

En este artículo se ha introducido mediante programación con *Scratch* una alternativa didáctica al trabajo con el sistema de coordenadas cartesianas. El planteamiento ha consistido en presentar de forma didáctica el tema tratado a la vez que introducir a los alumnos en la programación con *Scratch* de forma aplicada.

En el artículo se ha realizado una retrospectiva del pensamiento computacional como herramienta de construcción de la realidad, haciendo referencia a los principales exponentes de esta corriente.

Una vez presentadas diferentes formas de aproximarse al sistema de coordenadas cartesianas, se ha introducido una situación de aprendizaje basada en el escenario de *Scratch*, como área de trabajo del sistema de coordenadas

cartesianas, presentando casos prácticos de funciones de tipo lineal, afín y cuadrática, que se aprovechan para trabajar conceptos como el área de figuras planas, los ángulos asociados a una pendiente, el paralelismo o la geometría dinámica.

Referencias bibliográficas

- [1] Currículo educativo de Educación Primaria. Consejería de Educación y Universidades del Gobierno de Canarias.
<http://www.gobiernodecanarias.org/educacion/web/primaria/informacion/>
Última visita: diciembre 2017.
- [2] Bocconi, S.; Chiocciariello A., Dettori G.; Ferrari A.; Engelhardt K. (2016). *Developing Computational Thinking in Compulsory Education, Implications for Policy and Practice*. Publications Office of the European Union, Luxemburgo.
- [3] Wing J. (2010). Computational Thinking: What and Why? The Link, The magazine of Carnegie Mellon University's School of Computer Science, 1-6.
- [4] Forman G; Pufall P. B (1988). *Constructivism in the Computer Age*. New Jersey, Estados Unidos: Lawrence Erlbaum Associates
- [5] Papert S.; Harel I. (1991). *Constructionism*. New York, Estados Unidos: Ablex Publishing Corporation.
- [6] Marji, M. (2014). *Learn to Program with Scratch*. San Francisco, California: No Starch Press. pp. xvii, 1-9, 13-15.
- [7] Corralero, N. (2011). Scratch. Programación fácil para educación primaria y secundaria, *Revista Digital Sociedad de la Información*, 29, 1-10.
- [8] Suárez, P., Andrés, C., & Sarmiento Porras, R. E. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. *Revista mexicana de investigación educativa*, 20(65), 607-641.
- [9] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52, 60–67.

- [10] Vázquez-Cano, E., & Ferrer Delgado, D. (2015). La creación de videojuegos con scratch en educación secundaria. *Communication Papers*, 4(6), 63–73.
- [11] Valle, J. E. M., & Salgado, V. C. (2013). Pensamiento lógico matemático con scratch en nivel básico. *Revista vínculos*, 9(1), 87-95.
- [12] Pierce R. Coordenadas cartesianas. Disfruta las matemáticas.com:
<http://www.disfrutalasmatemáticas.com/graficos/coordenadascartesianas.html>. Última visita: diciembre 2017.
- [13] School Age: Sea Battle. Abcjuegos.net.
<http://www.abcjuegos.net/juego/school-a:ge-sea-battle>.
Última visita: diciembre 2017.
- [14] Santana, R (2015). Situación de Aprendizaje: El Cartesiano. Consejería de Educación y Universidades, Gobierno de Canarias. En <http://www3.gobiernodecanarias.org/medusa/ecoescuela/sa/2014/09/02/el-cartesiano/>
Última visita: diciembre 2017.
- [15] Creación de escenarios para Scratch a partir de imágenes existentes en gran formato. En <http://www.ambientepraxis.org/>
Última visita: diciembre 2017.
- [16] Castelnuovo E. (1981). *La Geometría*. Barcelona, España: Ketres.