

## CONCEPTOS DEL PENSAMIENTO COMPUTACIONAL

A medida que los jóvenes diseñan medios interactivos con Scratch, comienzan a interactuar con un conjunto de conceptos computacionales (esquema de los bloques para programar con Scratch), comunes a muchos lenguajes de programación. Hemos identificado siete conceptos que son muy útiles para una amplia gama de proyectos con Scratch y pueden transferirse a otros contextos ya sean estos de programación o no: *secuencias*, *ciclos*, *paralelismo*, *eventos*, *condicionales*, *operadores* y *datos*. Ofrecemos para cada concepto una definición y un ejemplo concreto de un proyecto de Scratch.

### Concepto: Secuencias

Un concepto clave en programación, es que una tarea o actividad particular se expresa como una serie de pasos o de instrucciones individuales, que puede ejecutar el computador. Tal como en una receta, una secuencia de instrucciones de programación indica el comportamiento o acción que se debe producir. Por ejemplo, el objeto gato puede programarse para que se mueva una distancia corta a través del escenario y diga, mediante la secuencia de instrucciones que se muestra en la Figura 1, “Yo estoy programando”.



FIGURA 1. Ejemplo de una secuencia de instrucciones

### Concepto: Ciclos

En el ejemplo anterior, se programó el gato para que se moviera 10 pasos, esperara 0.2 segundos y repitiera la acción, moviéndose otros 10 pasos y esperando otros 0.2 segundos. ¿Qué pasaría si en lugar de una repetición individual de la acción, quisiéramos que el gato se moviera y esperara tres veces más? Fácilmente, podríamos adicionar más bloques *mover* y *esperar*. ¿Pero si lo que queremos ahora es que el gato se mueva y espere 50, 100 o 1000 veces más? Los ciclos son mecanismos que ejecutan la misma secuencia, múltiples veces. La Figura 2 ilustra cómo, de manera más sucinta, se puede usar un ciclo para expresar una secuencia de instrucciones. En lugar de mover y esperar

usando 8 bloques consecutivos, vamos a usar tres bloques: *mover 10 pasos*, seguido por *esperar 0.2 segundos*, anidadas dentro de *repetir* con el número deseado de iteraciones.



FIGURA 2. Secuencia de instrucciones repetidas, expresadas como un ciclo

### Concepto: Eventos

Eventos, una cosa que desencadena que otra se suceda; es un componente esencial de los medios interactivos. Por ejemplo, el botón de inicio que activa la reproducción de un video musical o el choque de dos objetos que ocasiona el aumento del puntaje de un juego. La Figura 3 ilustra diferentes situaciones en las que un evento producirá una acción: (1) cuando se presiona la bandera verde el objeto dará vueltas continuamente con incrementos de 15 grados; (2) cuando se presiona la barra espaciadora el objeto se moverá hacia arriba y hacia abajo; y (3), cuando se hace clic sobre el objeto con el ratón, se desplegará durante 2 segundos una burbuja de texto que dice "¡Hola!".



FIGURA 3. Ejemplos de eventos que producen acciones.

### Concepto: Paralelismo

Los lenguajes de computador más modernos soportan paralelismo; esto es, secuencias de instrucciones que se suceden simultáneamente. Scratch permite paralelismo entre objetos. Por ejemplo, la escena del baile de una fiesta puede incluir varios personajes que bailan simultáneamente, cada uno con su propia secuencia de instrucciones de baile. Scratch permite también paralelismo dentro del mismo objeto. En la Figura 4, el gato de Scratch se ha programado para realizar, en paralelo, tres conjuntos de actividades como respuesta al evento "cuando se hace clic en la bandera verde": (1) suena continuamente una música de fondo; (2) continuamente baila hacia delante y hacia atrás; y (3), se presenta y habla de sus intereses.

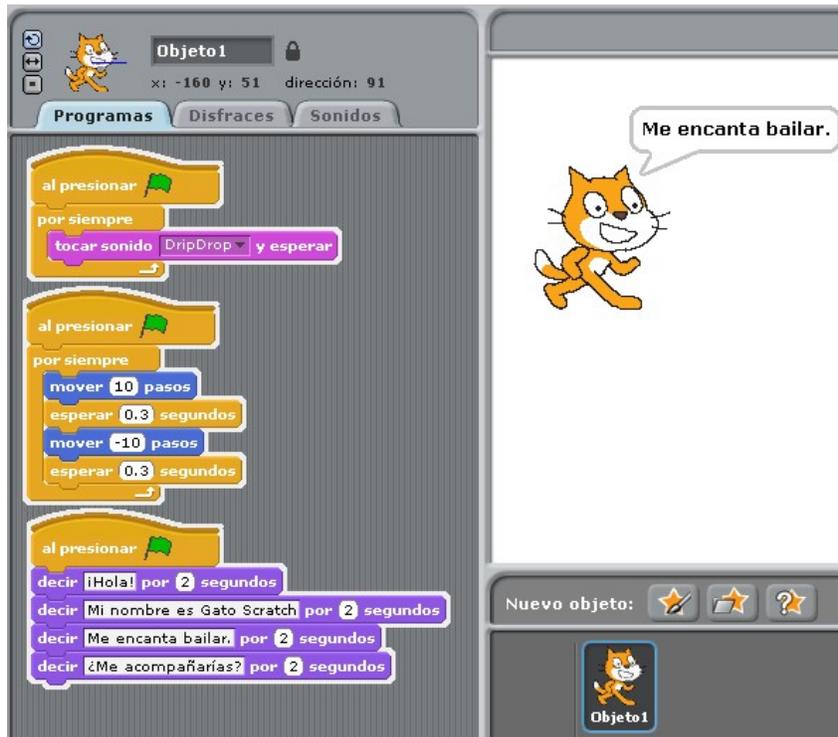


FIGURA 4. Ejemplo de paralelismo con un solo objeto.

**Concepto: Condicionales**

Otro concepto clave de los medios interactivos son los condicionales; esto es, la habilidad de tomar decisiones con base en ciertas condiciones, que apoyan la expresión de múltiples resultados. La Figura 5 ilustra el uso de un condicional; el bloque *si*, para determinar la visibilidad de un objeto. Si el cubo está *tocando el color amarillo*, debe comenzar a desvanecerse y reaparecer en el próximo nivel del juego; si no lo toca, debe permanecer visible.

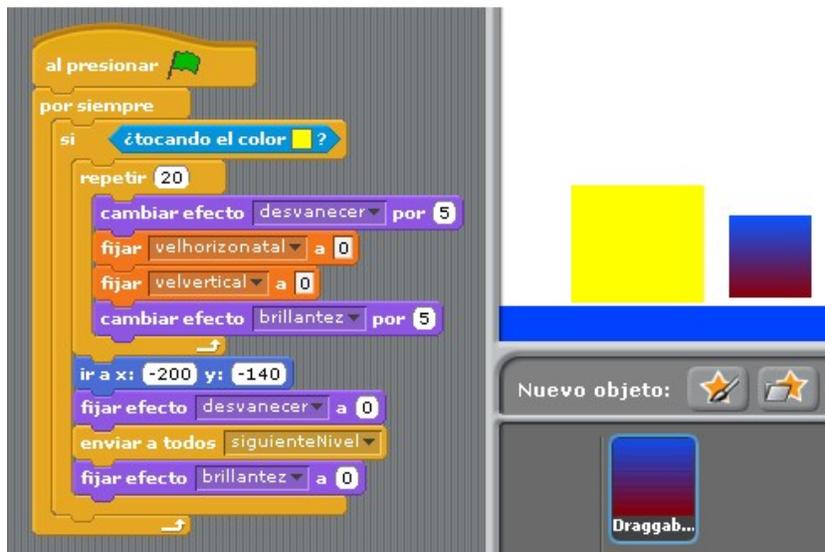


FIGURA 5. Ejemplo de condicionales.

## Concepto: Operadores

Los operadores ofrecen apoyo a las expresiones matemáticas, lógicas y de cadenas de caracteres, permitiendo al programador realizar manipulaciones numéricas y de cadenas. Scratch permite un rango de operaciones matemáticas que incluyen suma, resta, multiplicación, división y también, funciones como seno y potencia; además, operaciones con cadenas, incluyendo concatenación y longitud de las cadenas. La Figura 6 ilustra los bloques de operadores de Scratch.

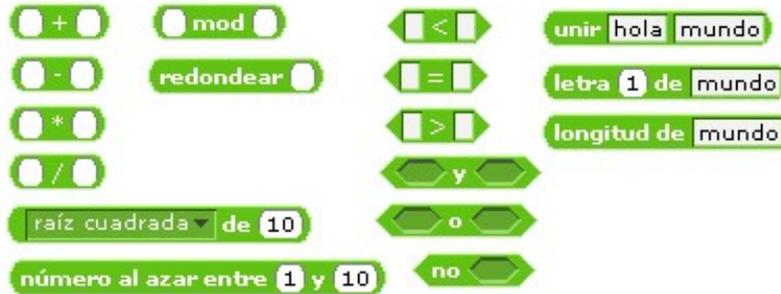


FIGURA 6. Bloques de operadores

## Concepto: Datos

Los datos incluyen guardar, recuperar y actualizar valores. Actualmente Scratch ofrece dos contenedores de datos: **variables** que pueden almacenar un solo número o cadena de caracteres y **listas** que pueden contener una colección de números o cadenas. Llevar el puntaje en un juego es, con frecuencia, un motivador para que los jóvenes diseñadores exploren las variables. La Figura 7 demuestra cómo se usa una variable para contabilizar el puntaje en un juego; por cada pescadito que se coma el pescado grande, el puntaje aumenta en 1.



FIGURA 7. Uso de variables para contabilizar puntajes.