

Trabajando con Scratch

Grado sugerido: Sexto

Josefina Blanco Quintero

Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.

Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](#)

Para contactar al autor/a de este recurso, escriba a: blancojosefina13@gmail.com

Manos a la obra. GUÍA 1. Trabajando con Scratch

| | |
|-------------------------------------|--|
| Aprendizajes esperados | <i>Con esta guía podrás alcanzar los siguientes aprendizajes:</i> <i>Identificar la secuencia de instrucciones como base de la programación</i> <i>Reconocer la importancia de los bucles en la programación</i> |
| Duración | <i>30 minutos</i> |
| Materiales Requeridos | <i>PC y guía virtual o impresa.</i> |
| Actividades para desarrollar | <i>Estas son las actividades necesarias para alcanzar los aprendizajes esperados:</i> <i>1. Conocer el entorno de Scratch</i> <i>2. Seguir las instrucciones que se detallan en la guía</i> |
| Adaptaciones | <i>Acá se brindan las sugerencias o recomendaciones para adaptaciones a diversos contextos.</i> <i>Scratch offline</i> |
| Referencias | <i>No aplica</i> |

Concepto

Los bucles son fragmentos de código que se ejecutan una y otra vez hasta que se cumpla una condición. Es una herramienta que permite automatizar tareas repetitivas.

La mecánica de la actividad consiste en hacer caminar al personaje cambiando su apariencia.

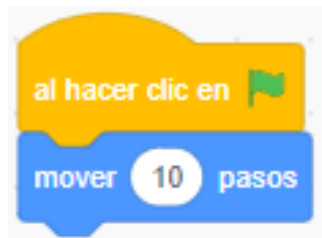
Para este primer ejercicio, utilizaremos los bloques de movimiento, apariencia, control y eventos.

Lo primero que vamos a hacer es observar el entorno de programación y ubicar los bloques que se van a utilizar. También, seleccionaremos el lenguaje Español (España) para identificar los bloques.

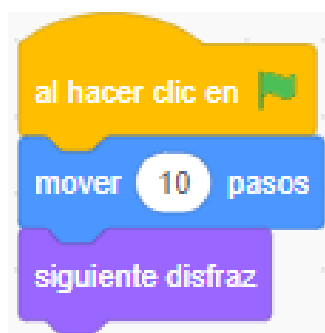
Nota: (Ejecutar el programa en cada paso para ir revisando los cambios)

Pasos.

1. Arrastrar el evento **al hacer clic en la bandera**
2. Arrastrar el bloque **mover 10 pasos** que se encuentra en la categoría movimiento



3. Arrastrar el bloque **siguiente disfraz** que se encuentra en la categoría de Apariencia

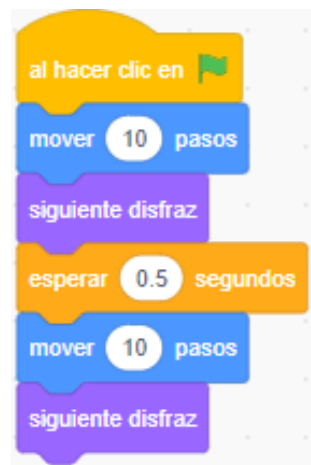


4. Repetimos los códigos que se encuentran dentro del evento para que se realice dos veces la acción.



¿Observas algún cambio?

Al ejecutar el código, no se logra observar el cambio ya que la acción sucede muy rápido. Para poder visualizar el evento, se necesita especificar el tiempo que dura cada acción. Para esto, arrastramos el bloque **“esperar__ segundos”** que se encuentra en la categoría de Control y lo ubicamos como se muestra la siguiente figura



Reto

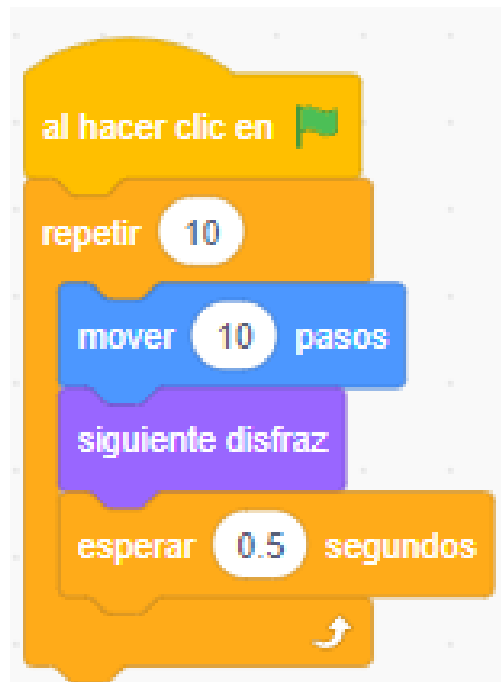
Organiza los bloques para que el gato pueda dar 7 pasos al presionar una sola vez la bandera.

Algunos seguramente pensarán en hacerlo de la siguiente manera:



¿Qué pasaría si tendrías que repetir esta acción 100 veces?

Para no extender los bloques de código y repetir las mismas acciones, se utilizan los bloques de control de flujo que te permiten especificar cuántas veces se va a repetir una o varias acciones, en programación se conocen como bucles. Para este ejemplo utilizaremos la estructura repetir.



¿Qué pudiste observar?

¿Podrías explicar lo que hace la estructura repetir?

Exploremos otros bucles



¿Qué diferencias encuentras entre los bucles vistos?

CIERRE

En la mayoría de los dispositivos que conocemos, encontramos patrones repetitivos llamados bucles que permiten automatizar acciones concretas, tal como la lavadora al ejecutar sus fases de lavado, el microondas midiendo el tiempo para calentar cierto alimento o el semáforo, alternando las luces para regular el tráfico; incluso en los juegos que tanto nos divierte. Los bucles están en todas partes, haciendo que la tecnología sea rápida, precisa.

Manos a la obra. FICHA 2. Coordenadas x,y / condicionales

| | |
|-------------------------------------|--|
| Aprendizajes esperados | <i>Con esta guía podrás alcanzar los siguientes aprendizajes:</i> <i>1. Utilizar el sistema de coordenadas x,y en actividades prácticas.</i> <i>2. Introducir conceptos básicos de programación como condicionales.</i> <i>3. Utilizar operadores aritméticos y relacionales.</i> |
| Duración | <i>40 minutos</i> |
| Materiales Requeridos | <i>PC y guía virtual o impresa.</i> |
| Actividades para desarrollar | <i>Estas son las actividades necesarias para alcanzar los aprendizajes esperados:</i> <i>1. Conocer el entorno de Scratch</i> <i>2. Tener los códigos de la guía 1</i> <i>3. Seguir las instrucciones que se detallan en la guía</i> |
| Adaptaciones | <i>Acá se brindan las sugerencias o recomendaciones para adaptaciones a diversos contextos.</i> <i>Scratch offline</i> |
| Referencias | <i>No aplica</i> |

Conceptos

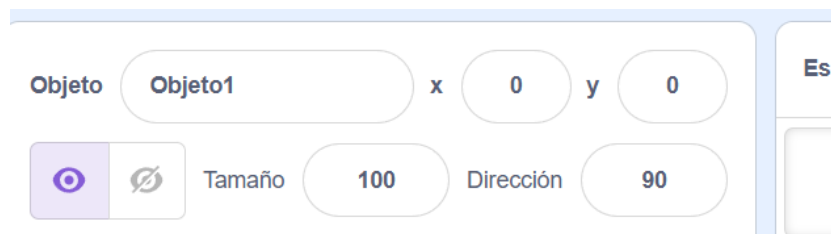
Coordenadas x,y: Son un par de valores que indican la posición de un punto en un plano

Condicionales: son herramientas que permiten a una secuencia de instrucciones tomar decisiones y ejecutar diferentes bloques de código según si una condición es verdadera o falsa, controlando así el flujo de ejecución.

En las fórmulas se utilizan símbolos especiales llamados operadores que, junto con datos y nombres de variables, forman combinaciones que se evalúan para obtener un valor final.

En este caso estaremos programando la rutina clásica de atrapar el objeto que se ve en muchos de los juegos que conocemos.

Invitamos a los estudiantes a realizar cambios en los valores de x y y que se encuentran en la parte inferior derecha. Se recuerda que las coordenadas pueden tomar valores negativos partiendo del punto de origen. Explicación sencilla de como estos valores pueden ser utilizados para programar movimiento

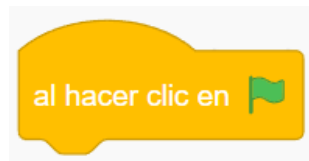


Pasos

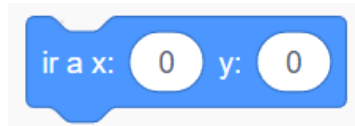
1. Invitamos a los estudiantes a que tomen un personaje de su agrado.

Nota: Es importante revisar las coordenadas en las que se encuentran los personajes ya que estas se necesitan para la programación.

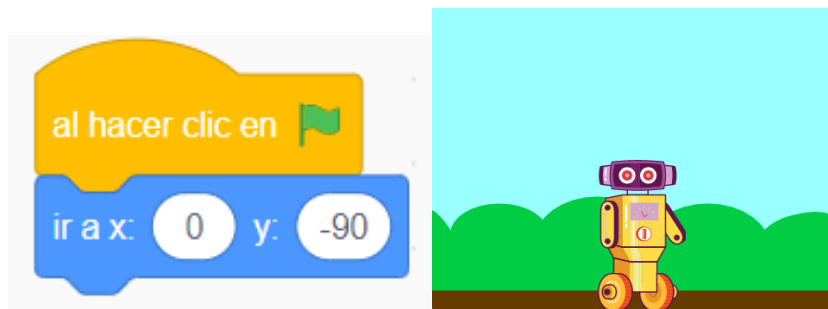
Arrastrar el evento al hacer clic en la bandera



y el bloque



El valor de **y** puede cambiar dependiendo del personaje.

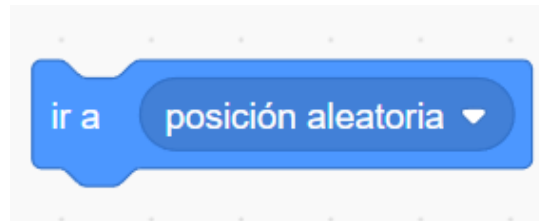


Se necesita que el personaje se mueva con el mouse de izquierda a derecha siempre.

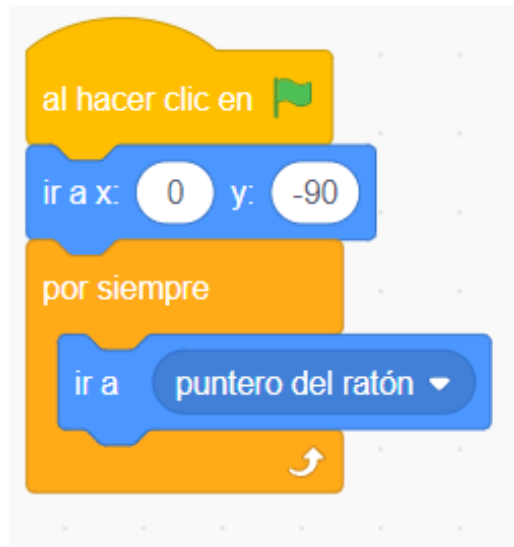
Como es "para siempre", agregamos un bucle por siempre



y la instrucción

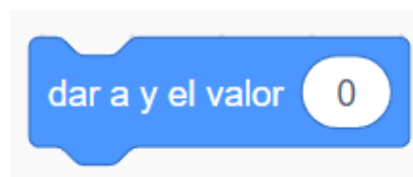


Desplegamos y seleccionamos puntero del ratón como se muestra en la figura

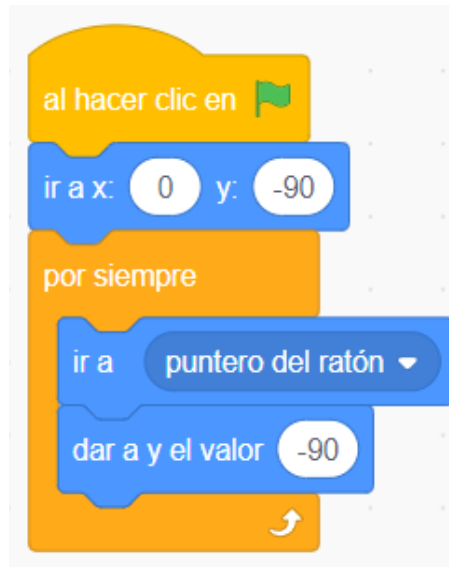


Ejecutamos para observar que hace...

3. Se observa que el personaje se mueve con el mouse en todas las direcciones. Para este ejemplo, se requiere que se mueva solo en la dirección x. Esto se logra fijando el valor de **y** con el bloque



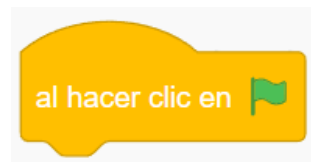
Escribimos -90 como se muestra en la figura



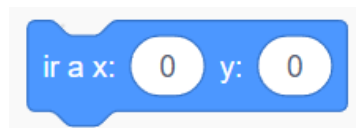
El bloque “dar a y el valor __” en otro lenguaje aparece como “fijar y a __”

Con esto se consigue que el movimiento con el mouse se de solo para la coordenada x.

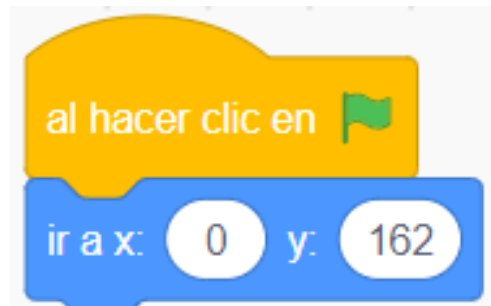
4. Agregamos el sprite que será atrapado por el robot. En este caso el balón. Teniendo seleccionado el objeto balón, arrastrar el evento al hacer clic en la bandera



y el bloque



El valor de **y** puede cambiar dependiendo del personaje. Lo importante es ubicarlo en el margen superior.

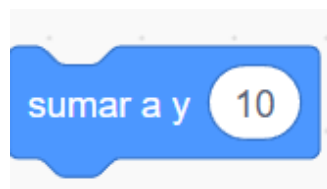


Como se necesita que el balón caiga (movimiento en posición y) se debe restar varias veces un valor a la posición actual para crear el efecto de caída.

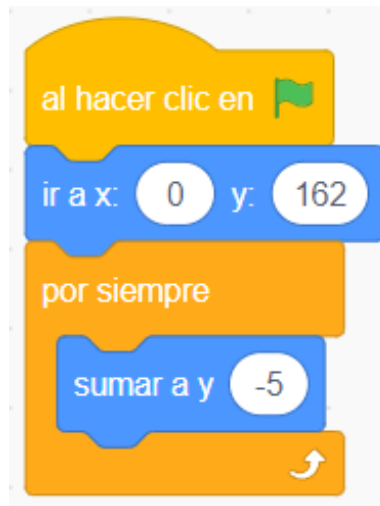
5. Agregamos un bucle por siempre para que se repita la acción de restar a posición actual.



y la instrucción



Cambiamos el valor como se muestra en la figura



El bloque “sumar a y ___” en otro lenguaje aparece como “cambiar y en ___”

6. Se necesita que el balón al alcanzar una posición en y (En este caso el suelo) vuelva al inicio para crear el efecto de caída para siempre.

Para esto, agregamos un condicional para que el movimiento del balón vuelva al inicio y cree el efecto de caída



Nota: Hay que tomar valores que acepte el bloque, es decir en la expresión lógica del condicional prestar atención con el valor de y que no puede ser muy pequeño porque no surte efecto. La posición es donde el balón se vea en su totalidad.

7. Agregamos el bloque “**ir a: posición aleatoria**” como se muestra en la figura para que el balón aparezca aleatoriamente.



Ejecuta el código.

¿Qué le podrías agregar al código?

CIERRE

Hoy nos sumergimos en el universo de las coordenadas (x,y). Desplazamos elementos con exactitud, aplicamos condicionales para tomar elecciones tal como un semáforo que decide entre seguir y parar. También manipulamos operadores para agregar, quitar o medir cantidades. ¿Se acuerdan cómo logramos que nuestro personaje virtual respondiera al alcanzar el límite o modificara su rumbo? Esas eran las bases: coordenadas, condicionales y operadores trabajando en sincronía.

Manos a la obra. FICHA 3. Variables

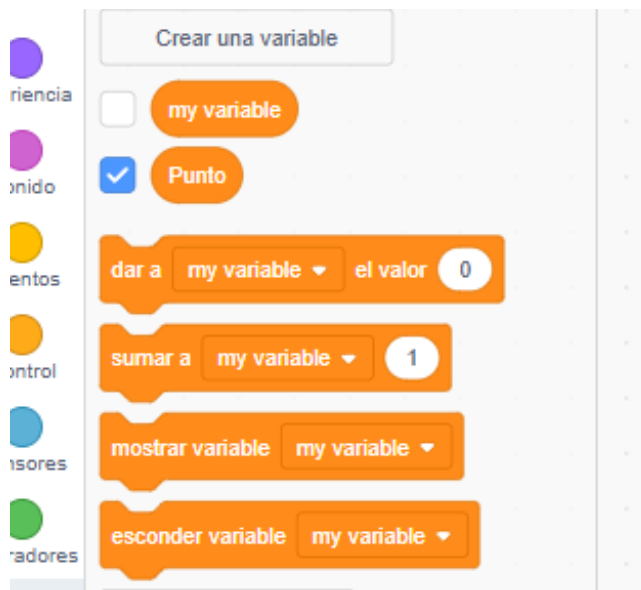
| | |
|-------------------------------------|---|
| Aprendizajes esperados | <i>Con esta guía podrás alcanzar los siguientes aprendizajes:</i> <i>1. Introducir el concepto de variable</i> |
| Duración | <i>45 minutos</i> |
| Materiales Requeridos | <i>PC y guía virtual o impresa.</i> |
| Actividades para desarrollar | <i>Estas son las actividades necesarias para alcanzar los aprendizajes esperados:</i> <i>1. Conocer el entorno de Scratch</i> <i>2. Tener los códigos de la guía 1,2</i> <i>3. Seguir las instrucciones que se detallan en la guía</i> |
| Adaptaciones | <i>Acá se brindan las sugerencias o recomendaciones para adaptaciones a diversos contextos.</i> <i>Scratch offline</i> |
| Referencias | <i>No aplica</i> |

Concepto

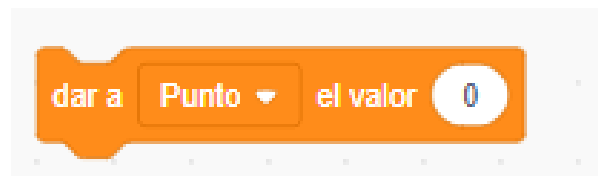
Las variables actúan como cajas que almacenan información que puede ser procesada en algún momento del código

Siguiendo con la actividad anterior, actualizaremos la versión para que al colisionar los personajes, se obtengan puntos.

1. En la categoría variable, crea una variable llamada punto.



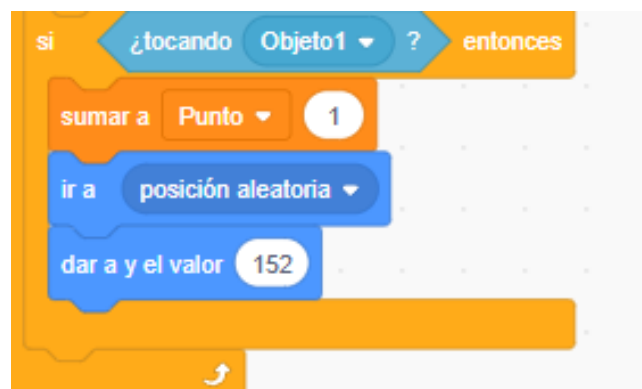
2. Agregar a la programación anterior del objeto balón el bloque



que se encuentra en la categoría variable como se muestra en la imagen



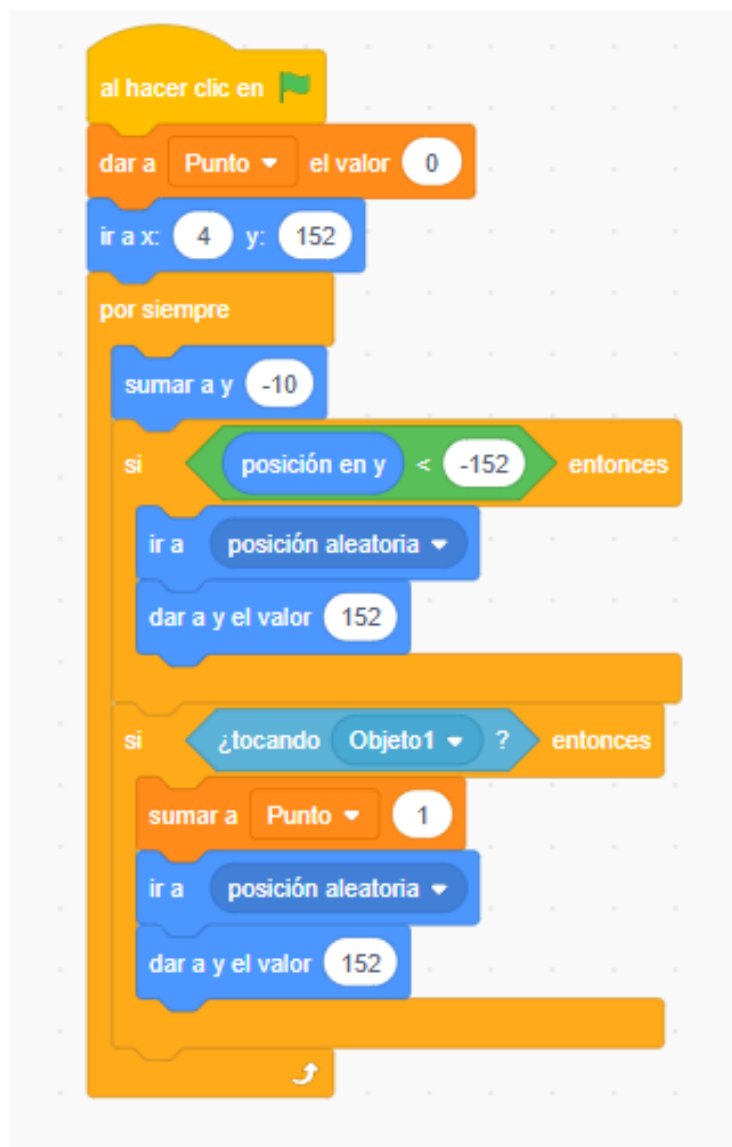
3. Necesitamos que cuando la pelota colisione con el personaje, se incremente en 1 el valor que se encuentra almacenado en la variable. Esto se puede conseguir agregando un condicional de la siguiente manera:



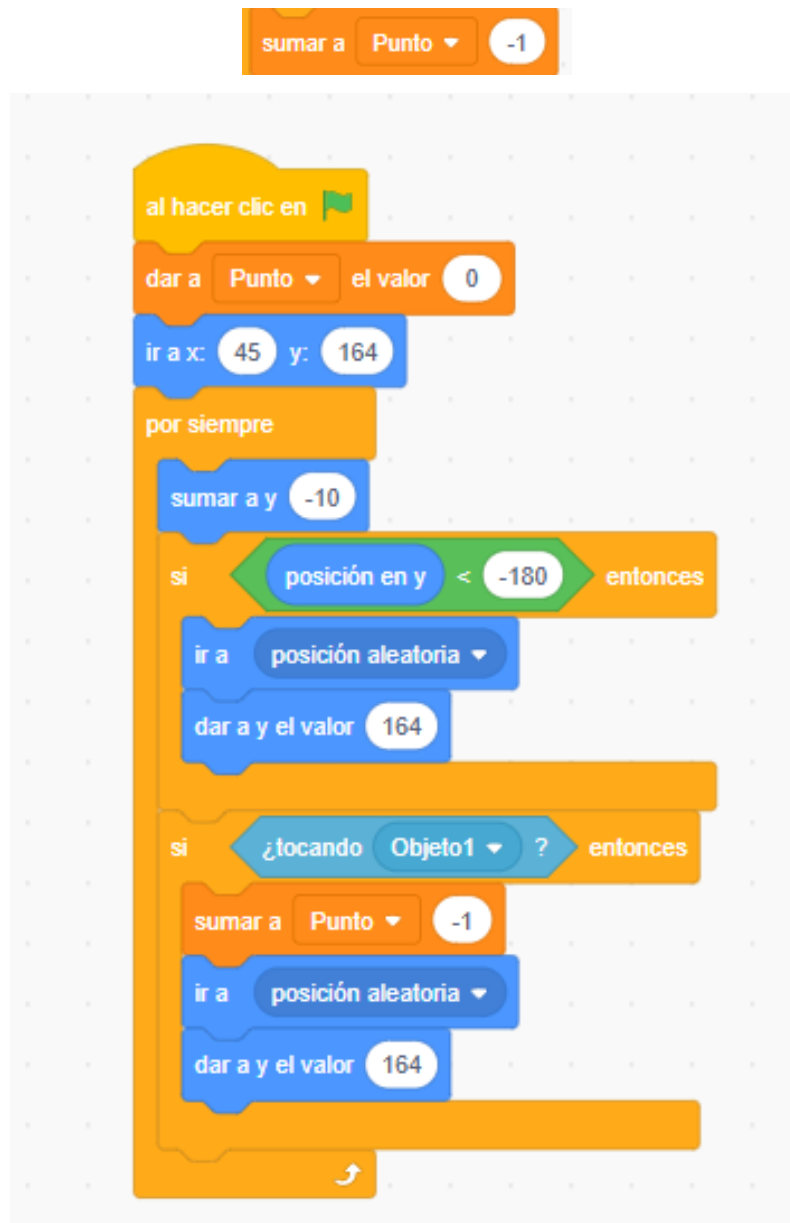
El bloque tocando objeto1 se encuentra en la categoría de sensores



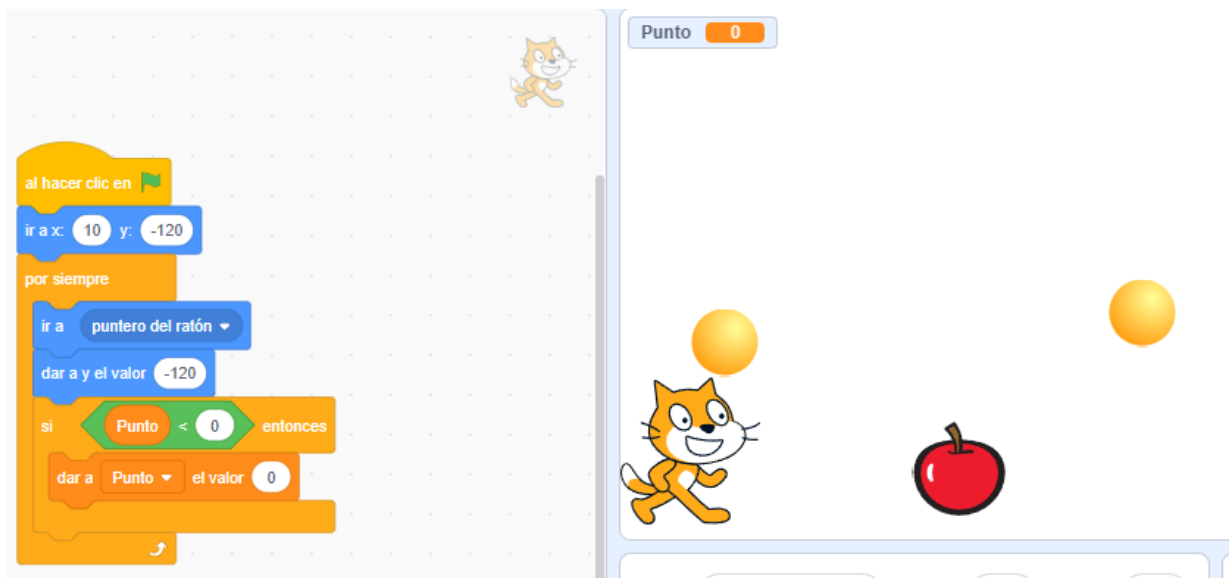
Las instrucciones completas se muestran en la siguiente figura:



4. Ahora, agregaremos un sprite que el personaje principal debe esquivar porque perdería puntos al colisionar con él. La configuración sería la misma que para el sprite anterior, con la diferencia que se sumaría uno negativo (-1) para que se reste puntos al tocarlo.



5. Por último, modifica los bloques del personaje principal añadiendo un condicional para que los puntos no sean menor a cero.



Reto

¿Podrías anexas otros condicionales como por ejemplo cambiar de velocidad al llegar a cierta cantidad de puntos, cambiar de escenario dependiendo de una condición o que el personaje cambie su apariencia unos instantes cuando pierda puntos?

CIERRE

Hoy descubrimos que una variable es como una caja en programación: puede guardar valores (como tus puntos en el juego) y cambiar cuando sea necesario. Por ejemplo, cada vez que ganabas puntos, la variable puntaje se actualizaba. Sin variables, los programas no podrían recordar ni calcular nada.