



Manual de Introducción a la Programación

*Red Científica Adain URO 2193
Región del Libertador General Bernardo O'Higgins, Chile*



Créditos

Edición:

Ana Paula Yáñez, Coordinadora Red Científica URO 2193.

Autores:

Ignacio Bugueño, Ingeniero Civil Eléctrico, docente e investigador.

Rancagua, Región del Libertador Bernardo O'Higgins, 2023.

Sobre el Manual de Introducción a la Programación

El presente manual presenta un glosario y una serie de actividades paso a paso que permitirán instalar el software requerido para las Capacitaciones en Introducción a la Programación y Robótica Educativa de Red Científica Adain URO 2193.

Esta guía considera el uso de Windows, Mac o otro sistema operativo de distribución Linux. Se compone de tres tutoriales: Instalación del Software, Glosario de Bloques de Instrucción, y Actividades.

¿Qué es Scratch?

Es un lenguaje de programación visual creado por el MIT (Instituto Tecnológico de Massachusetts) y orientado a facilitar su aprendizaje de forma intuitiva mediante bloques. ¿Y esto qué significa?

“Que en vez de escribir código vamos a tener diferentes piezas como si se tratase de un rompecabezas.”

Iremos encajando bloques para desarrollar las diferentes instrucciones hasta tener nuestro proyecto, que podría ser perfectamente una historieta, una animación, un videojuego o un robot inteligente.

Es decir, utilizamos Scratch. para construir scripts juntando bloques que representan órdenes, instrucciones de control, expresiones, etc. El conjunto de bloques de Scratch se dividen en diferentes categorías, los bloques de cada categoría se distinguen por un color diferente.

Cada personaje y escenario puede tener sus propios códigos. El conjunto de bloques a partir de los cuales se pueden construir códigos para un escenario es ligeramente diferente del utilizado para los personajes, por ejemplo la categoría Movimiento para el escenario está vacía, porque el escenario no se puede mover ni girar. En primer lugar, hablaremos de los distintos tipos de bloques para sprites y, a continuación, se proporcionará información básica sobre los bloques disponibles para el escenario.

¡Bienvenidos a la programación divertida!





Guía de Instalación de Scratch en Sistema Operativo Windows

1.1 Instalación de Scratch en Windows

Es un lenguaje de programación visual creado por el MIT (Instituto Tecnológico de Massachusetts) y orientado a facilitar su aprendizaje de forma intuitiva mediante bloques.

Para su instalación se deben seguir los siguientes pasos:

1. Entrar a <https://scratch.mit.edu/download>
2. Seleccione su sistema operativo

Elija su sistema operativo:

 Windows

 macOS

 ChromeOS

 Android

3. Descienda en el sitio web y le aparecerán dos opciones: “Obtén la app de Scratch en la Tiendas de Microsoft” o “**Descarga Directa**”. Recomendamos la segunda opción



4. Una vez descargado el archivo, ejecutar el archivo descargado.
5. Seguir con la configuración recomendada hasta que la instalación esté finalizada.
6. ¡Listo!

1.2 Verificar instalación en Windows

Para verificar la correcta instalación de Scratch se deben seguir los siguientes pasos:

1. Buscar el ícono de Scratch en su Escritorio



2. Hacer doble click en él
3. Ejecutar Scratch

Al ejecutar Scratch, aparecerá un software similar al siguiente:

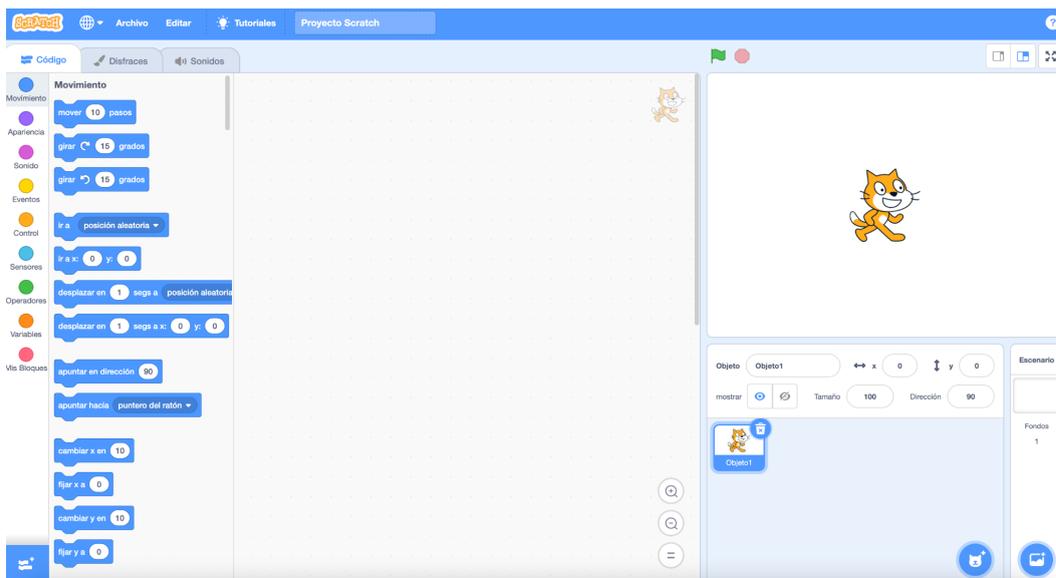


Figura: Visualización de Programa Scratch

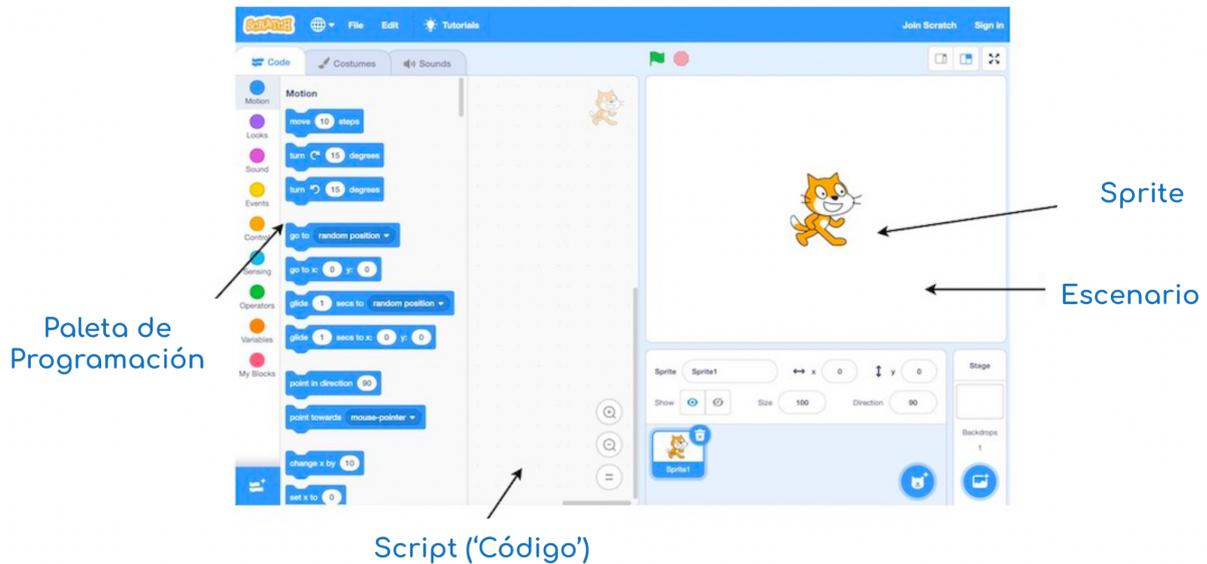
¡El software ya está instalado y listo para empezar con las siguientes actividades!



Fundamentos básicos de Scratch

1.1 Vista global de Scratch

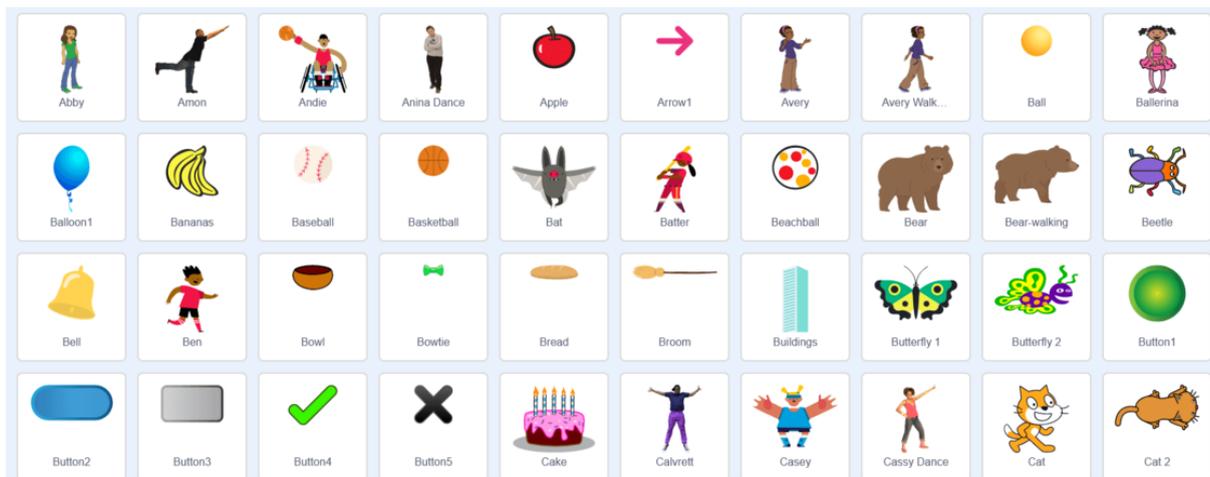
El panel principal de scratch se compone de cuatro zonas importantes: el sprite, el escenario, la paleta de programación, y el script (código a implementar), tal como se muestra a continuación



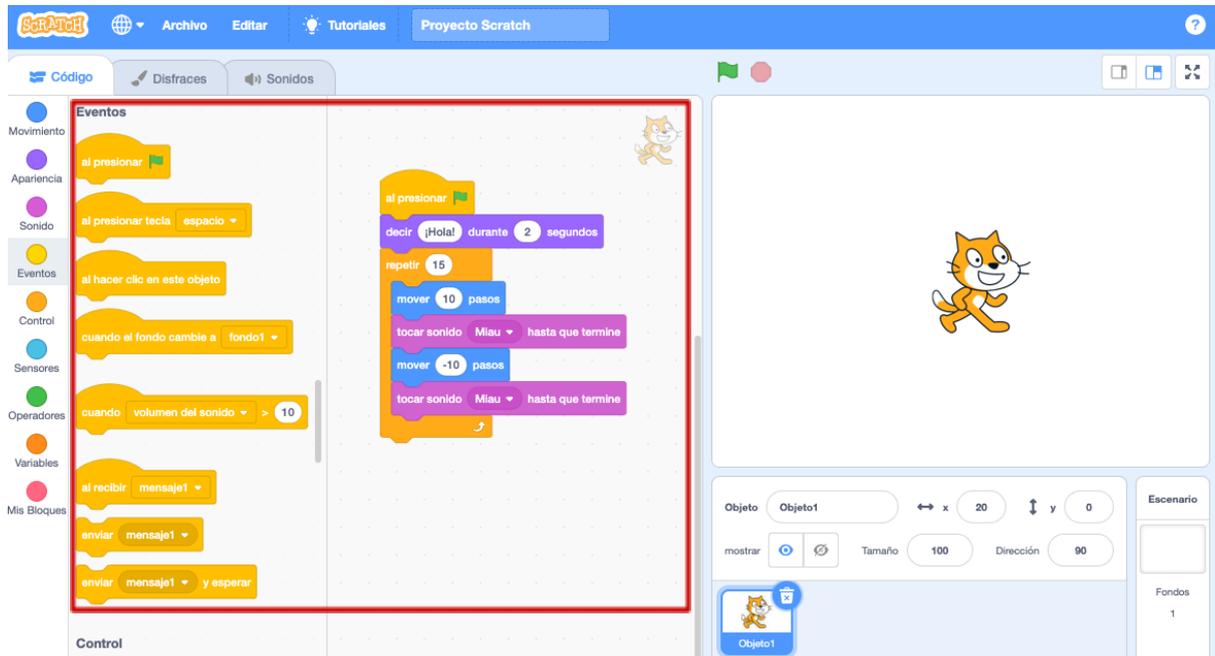
1.2 Sprites (Personajes)

Los sprites son la parte principal del programa. Son los personajes o actores de la pantalla.

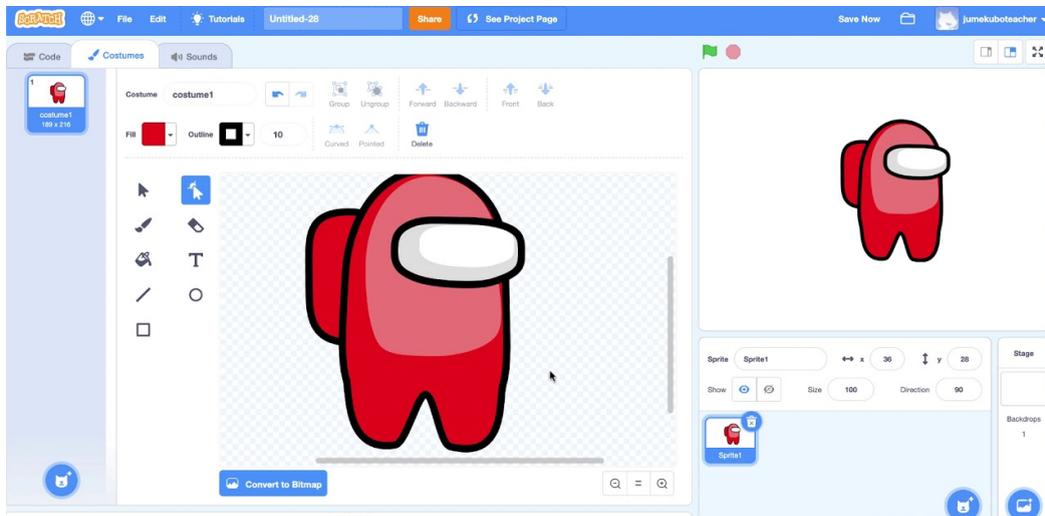
El sprite más básico es un gato naranja y blanco, pero también hay muchos otros. Están los sprites de animales, personas y dibujos animados que hacen todo el movimiento. También hay instrumentos musicales, pelotas, comida ropa y sprites de letras que no deben moverse.



Los sprites pueden estar en cualquier lugar de la pantalla y los sprites que se mueven pueden desplazarse utilizando bloques de movimiento

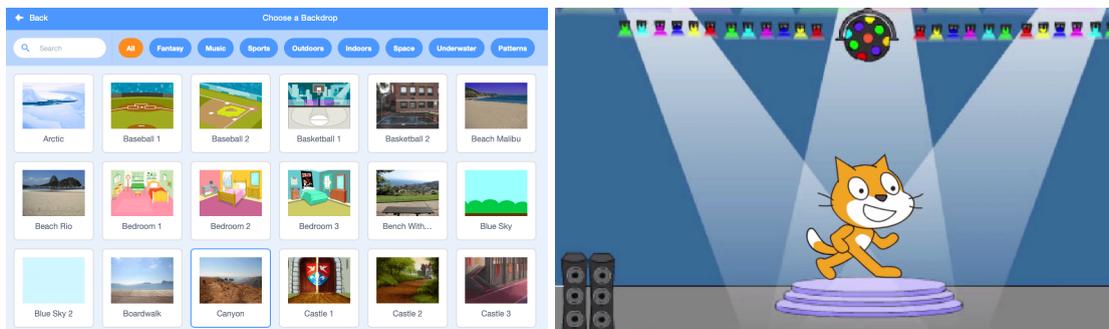


Inclusive, ¡si no encuentras el sprite que quieres, puedes dibujarlo!



1.3 Escenarios

Los escenarios están en el fondo de la pantalla. Son el ambiente de la escena. Los escenarios pueden ser coloridos, aburridos, espeluznantes, divertidos, etc. Los sprites están en el escenario.



Al elegir un sprite para el fondo, este debe encajar con el escenario. Ejemplo: un cangrejo quedaría mejor sobre un fondo marítimo que en un estadio.



1.4 Paleta de programación & Script

Es la zona donde se implementa el comportamiento del personaje, de los objetos y del escenario. Aquí se arrastrán y ejecutarán los diversos bloques existentes.



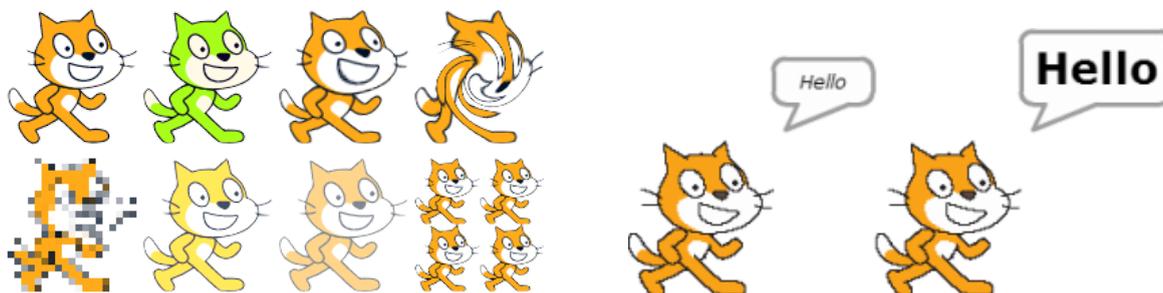
1.5 Bloques de Movimiento

¡Estos bloques mueven el sprite! Todos los movimientos que necesites los encontrarás aquí (a menos que no exista)



1.6 Bloques de Apariencia

¡En estos bloques puedes hacer que tu sprite 'hable'! También puedes cambiar su apariencia.



1.7 Bloques de Sonido

Aunque resulte redundante, ¡estos bloques emiten sonido predefinidos! Como por ejemplo, sonidos de instrumentos.



1.8 Bloques de Eventos

Los bloques de eventos ejecutan el código. Sólo van al inicio. Indican el comienzo y cuando el sprite debe escuchar el código.



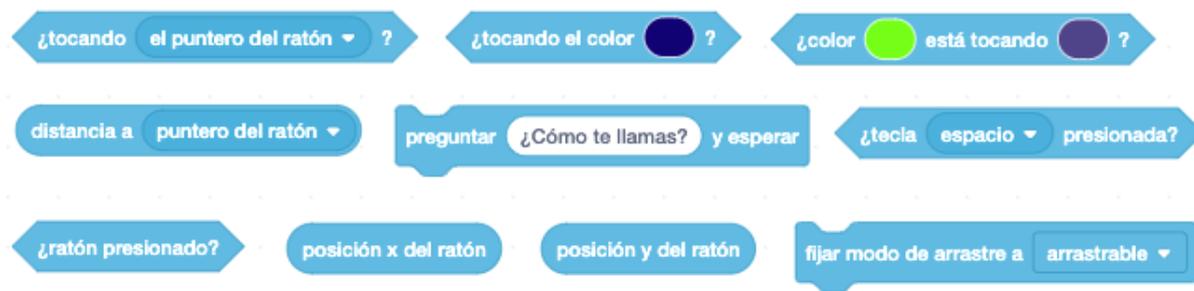
1.9 Bloques de Control

En estos bloques se encuentran las sentencias if then y los bucles. Puede que no sean la única razón por la que tu sprite hace cosas, pero son una gran parte de la razón por la que tu sprite hace cosas.



1.10 Bloques de Sensado

Detecta colores o si está tocando un borde o el puntero del ratón. Preguntar y esperar respuesta. Establecer fecha, respuesta, temporizador y nombre de usuario



1.11 Bloques de Operación

Bloques que incluyen el cálculo de enteros u otras variables, valor absoluto y módulos.



1.12 Bloques de variable

Hacer, establecer, cambiar, mostrar y ocultar una variable.





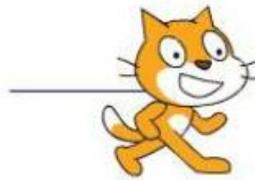
Glosario de bloques básicos de Scratch

1. Movimiento

La categoría **Movimiento** agrupa bloques de movimiento de los personajes, utilizados para su movimiento, rotación y lectura de datos actuales relacionados con la ubicación de un personaje. Se trata de bloques básicos muy utilizados. A continuación se describen todos los bloques de esta categoría.



Mueve el personaje una cierta distancia de acuerdo con la dirección actual. El valor por defecto puede editarse. Si queremos mover el personaje hacia atrás, debemos introducir un valor negativo.



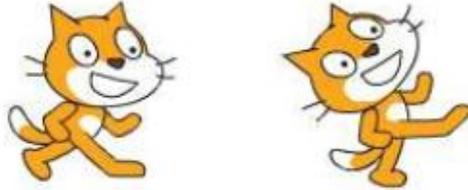
Rota al personaje hacia la derecha (en el sentido de las agujas del reloj) un grado especificado. El valor por defecto puede editarse.



girar ↶ 15 grados

1.3

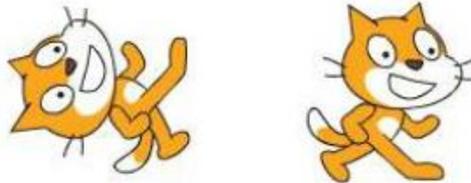
Rota el personaje a la izquierda (en sentido contrario a las agujas del reloj) un grado especificado. El valor por defecto puede ser editado. Esto es equivalente a una rotación a la derecha (sentido horario) con un valor negativo.



apuntar en dirección 90

1.4

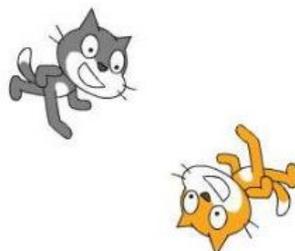
Fija el sprite en una dirección determinada. Podemos elegir una de las cuatro direcciones de la lista (90 grados - derecha, -90 grados - izquierda, 0 grados - arriba, 180 grados - abajo) o introducir manualmente un ángulo específico en grados.



apuntar hacia puntero del ratón ▾

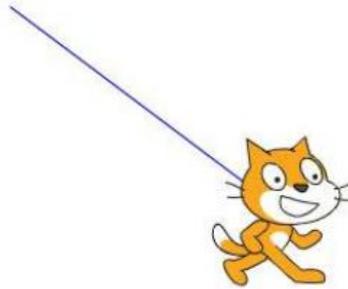
1.5

Establece el sprite en la dirección del cursor del ratón o de otro sprite. El objeto (un sprite específico o el cursor del ratón) debe estar seleccionado en la lista desplegable.

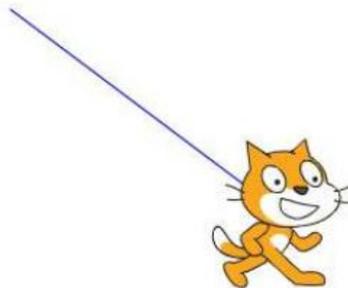




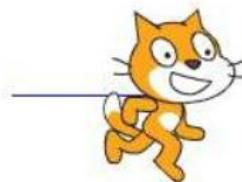
Mueve el sprite a un punto con coordenadas específicas (x, y) de acuerdo con la dirección actual y la configuración actual del lápiz (aplicado/levantado, color, grosor).



Similar al comando **ir a x: y:**, pero ejecutado durante un periodo de tiempo especificado.

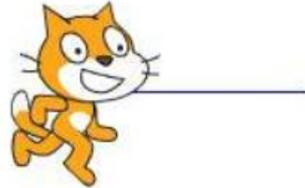


Cambia la coordenada x del sprite por un valor específico de acuerdo con la dirección actual y la configuración actual del lápiz. El valor por defecto puede editarse.



1.9 fijar x a 0

Mueve el sprite a un punto con un valor de coordenada x específico (el valor de la coordenada y permanece inalterado) de acuerdo con la dirección actual y la configuración actual del lápiz. El valor por defecto puede editarse.



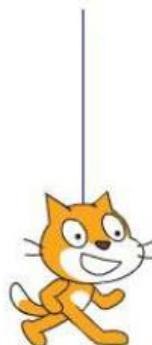
1.10 cambiar y en 10

Cambia la coordenada y del sprite por un valor específico de acuerdo con la dirección actual y la configuración actual del lápiz. El valor por defecto puede editarse.



1.11 fijar y a 0

Mueve el sprite a un punto con un valor de coordenada y específico (el valor de la coordenada x permanece inalterado) de acuerdo con la dirección actual y la configuración actual del lápiz. El valor por defecto puede editarse.



si toca un borde, rebotar

1.12

Cambia la dirección a la opuesta cuando el sprite toca el borde del escenario.

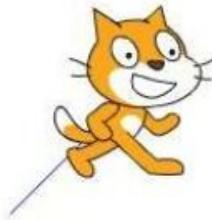
fijar estilo de rotación a izquierda-derecha ▾

1.13

Especifica el comportamiento del sprite en función de la dirección. Hay tres opciones en la lista desplegable que podemos elegir:

- en todas las direcciones: el sprite mira según su dirección,
- izquierda-derecha: el sprite mira sólo a la izquierda o a la derecha,
- no rotar: el sprite no reacciona al cambio de dirección.

Independientemente de la forma de indicar la dirección, el sprite se mueve de acuerdo con su dirección.



posición en x

1.14

El valor actual de la coordenada x del sprite, es decir: su centro con respecto al escenario. El valor devuelto puede utilizarse como argumento en expresiones. Al marcar la casilla se muestran las coordenadas actuales en el escenario. Adopta los valores comprendidos entre -240 y 240.

Objeto1: posición x 0



1.15

posición en y

El valor actual de la coordenada y del sprite, es decir: su centro con respecto al escenario. El valor devuelto puede utilizarse como argumento en expresiones. Al marcar la casilla se muestran las coordenadas actuales en el escenario. Adopta los valores comprendidos entre -180 y 180.

Objeto1: posición y 0



1.16

dirección

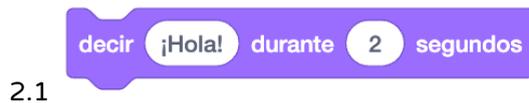
Valor actual de la dirección (orientación) del sprite. El valor devuelto puede utilizarse como argumento en expresiones. Marcando la casilla se muestra la dirección actual en el escenario. Adopta los valores comprendidos entre -180 y 180 grados.

Objeto1: dirección 90



2. Apariencias

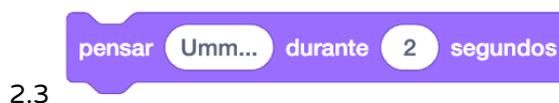
La categoría **Apariencia** es una colección de bloques de apariencia de sprites relativos al vestuario, tamaño, visibilidad, etc. Algunos de estos bloques se utilizan con mucha frecuencia. A continuación encontrará una descripción de la mayoría de los bloques de esta categoría.



Durante un tiempo limitado muestra en un globo de diálogo el texto mostrado en el cuadro de edición. Esto significa detener la ejecución del script durante un número determinado de segundos. También se puede arrastrar una variable al cuadro de edición o construir una expresión en él - a menudo para conectar la información mostrada utilizamos el bloque join.



Muestra en una burbuja de diálogo el texto que aparece en el cuadro de edición y no detiene la ejecución del script. En consecuencia, se ejecutan más instrucciones, lo que podría provocar la rápida desaparición de la burbuja. También se puede arrastrar una variable al cuadro de edición o construir una expresión en él - a menudo para conectar la información mostrada utilizamos el bloque join.



Durante un tiempo limitado muestra en una burbuja de pensamiento el texto mostrado en el cuadro de edición. Esto significa detener la ejecución del script durante un número determinado de segundos. Funciona de la misma manera que el comando say, pero se muestra un tipo de información diferente.

pensar Umm...

2.4

Muestra en una burbuja de pensamiento el texto mostrado en el cuadro de edición y no detiene la ejecución del script. En consecuencia, se pueden ejecutar inmediatamente otras instrucciones, lo que podría provocar la rápida desaparición de la información. Funciona de la misma manera que el comando **decir**, sólo que es un tipo diferente de burbuja. También se puede arrastrar una variable al cuadro de edición o construir una expresión en él.



mostrar

2.5

Muestra la imagen de un sprite (el traje actual) si el sprite está oculto. Si el sprite está visible, no tiene ningún efecto.

esconder

2.6

Oculto la imagen de un sprite si éste es visible. Si el sprite es invisible, no tiene ningún efecto.

cambiar disfraz a disfraz2 ▾

2.7

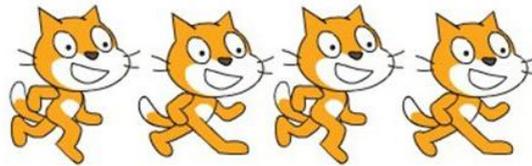
Hace que el sprite cambie su traje a uno específico, seleccionado de la lista desplegable. Por defecto, el sprite (gatito) tiene dos trajes.



siguiente disfraz

2.8

Hace que el sprite cambie su traje por uno posterior. El siguiente traje después del último es de nuevo el primero de la lista. Este bloque se utiliza a menudo para la animación de sprites.



cambiar fondo a fondo1

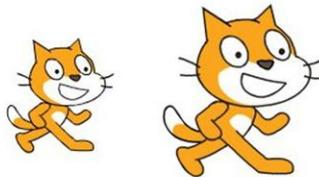
2.9

El escenario puede constar de varios escenarios (por defecto sólo hay uno, de fondo blanco). Con este bloque el sprite puede cambiar el fondo a uno seleccionado de la lista desplegable (con un nombre específico, el siguiente fondo, el fondo anterior).

cambiar tamaño en 10

2.10

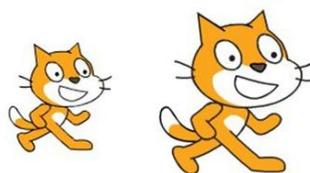
Se utiliza para cambiar el tamaño del sprite (todos los trajes) en un valor especificado. Si queremos reducir el tamaño del sprite, debemos introducir un argumento negativo.



fijar tamaño a 100 %

2.11

Se utiliza para cambiar el tamaño del sprite (todos los trajes) a un valor porcentual especificado. Si queremos restaurarlo a su tamaño original, el tamaño debe establecerse en 100%.



ir a la capa de adelante ▾

2.12

Si el sprite está cubierto por otros sprites, se mostrará en la parte superior.



disfraz número ▾

2.13

El número actual del traje del sprite. Puede usarse como argumento en expresiones. Al marcar la casilla se muestra el número actual en el escenario.

Objeto1: número de disfraz 1



fondo número ▾

2.14

Nombre del fondo mostrado actualmente. Puede utilizarse como argumento en expresiones. Al marcar la casilla se muestra el nombre del fondo en el escenario.

número de fondo 1



2.15 tamaño

Tamaño actual del sprite expresado como valor porcentual. Puede utilizarse como argumento en expresiones. Al marcar la casilla se muestra el tamaño en el escenario.

Objeto1: tamaño 100



3. Sonidos

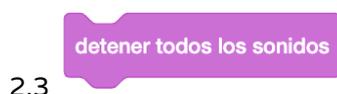
La categoría **Sonido** es un grupo de bloques para sonidos, instrumentos y notas musicales. A continuación se describen sólo los bloques de reproducción de archivos de audio. Un personaje puede estar conectado con varios sonidos (archivos de audio asignados), por defecto el sprite gato tiene asignado un sonido - miau. Para cambiar los sonidos asignados a un sprite dado (quitar/añadir un archivo de sonido, grabar nuestro propio audio) usamos la pestaña Sonidos.



Reproduce el archivo de audio seleccionado en la lista desplegable. Las instrucciones de script subsiguientes se ejecutan independientemente del audio que se esté reproduciendo.



Hace que se reproduzca el archivo de audio seleccionado en la lista desplegable y pausa la ejecución del script. Las instrucciones de script posteriores se ejecutan una vez finalizada la reproducción del archivo de audio.



Detiene todos los sonidos reproducidos actualmente.

4. Variables

La categoría **Variables** contiene botones para crear variables. Una vez que hayamos creado nuestras propias variables, aparecerán bloques para modificar y utilizar valores. Las variables nos permiten **almacenar** datos. Scratch también utiliza variables, por ejemplo en la categoría Movimiento podemos obtener el valor de las coordenadas de la posición actual del sprite o de su dirección.

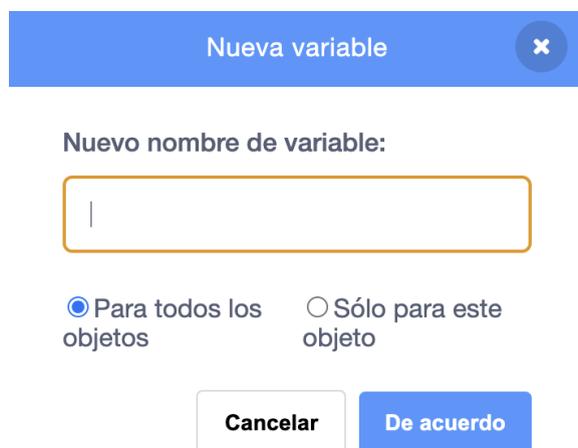
4.1 Creando una variable

Si no hemos creado ninguna variable propia, habrá un botón disponible en la categoría Variable.



Utilizaremos variables sólo para almacenar datos numéricos.

Después de hacer clic en el botón **Crear una variable**, aparecerá un cuadro de diálogo en la pantalla, como se muestra a continuación.

A dialog box titled "Nueva variable" with a close button (X) in the top right corner. Below the title is a label "Nuevo nombre de variable:" followed by a text input field containing a vertical bar cursor. Underneath are two radio button options: "Para todos los objetos" (selected) and "Sólo para este objeto". At the bottom are two buttons: "Cancelar" and "De acuerdo".

Nueva variable ✕

Nuevo nombre de variable:

Para todos los
objetos

Sólo para este
objeto

Cancelar

De acuerdo

Tenemos que especificar el nombre de la variable y seleccionar su disponibilidad: si la variable debe estar disponible para todos los sprites (también para el escenario) o sólo para un sprite específico (en este caso otros sprites no pueden ver esa variable; también pueden tener una variable con el mismo nombre, pero serán variables diferentes).

Después de confirmar con el botón **De acuerdo** aparecerán bloques para habilitar las operaciones sobre las variables.

4.2 contador

Valor actual de la variable. Puede utilizarse como argumento en expresiones. Al marcar la casilla se muestra el valor actual de una variable en el escenario.

contador 0



Cuando hacemos click con el botón derecho del ratón en el bloque con el nombre de la variable en el área de datos, podemos seleccionar una de las dos opciones del menú contextual: cambiar el nombre de una variable o eliminarla.

Renombrar variable
Eliminar la variable "contador"

4.3 fijar contador a 0

Asigna el valor de la variable. Podemos editar el valor por defecto, también podemos insertar una expresión en su lugar.

4.4 cambiar contador en 1

Cambia el valor de una variable por un valor especificado.

4.5 mostrar variable contador

Muestra el valor de la variable en el escenario. Funciona de la misma forma que cuando se edita la casilla de verificación del nombre de la variable.

4.6 esconder variable contador

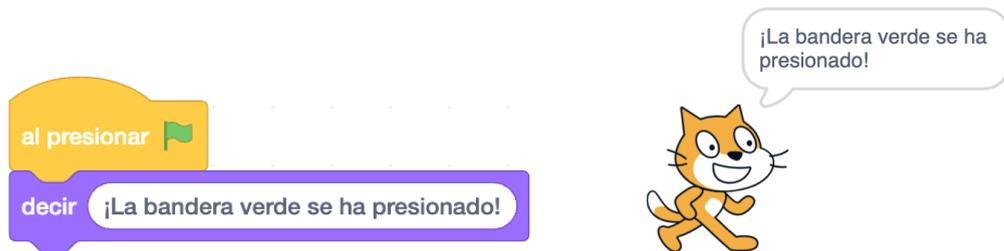
Oculto la visualización del valor de la variable en el escenario. Funciona de la misma manera que cuando editamos la casilla del nombre de la variable para desactivarla.

5. Eventos

La categoría Eventos incluye bloques que inician scripts que reaccionan a eventos específicos (por ejemplo, hacer click en un sprite con el ratón, pulsar una tecla del teclado) y relacionados con el soporte de mensajes (eventos autogenerados en el programa). A continuación se describen casi todos los bloques de esta categoría.



El bloque básico. Inicia la mayoría de los scripts. Los scripts se ejecutan cuando hacemos click en la bandera verde de la pantalla: iniciar el programa.



Ejecuta el script después de pulsar una tecla específica del teclado. Seleccionamos una tecla de una lista desplegable.



5.3

al hacer clic en este objeto

El bloque ejecuta el script cuando hacemos click con el botón izquierdo del mouse en el sprite.



5.4

cuando el fondo cambie a fondo1 ▾

El bloque ejecuta el script cuando se produce un cambio de fondo. El nombre del fondo puede seleccionarse en una lista desplegable. El script se iniciará cuando el fondo se convierta en el actual (mostrado en el escenario).



5.5

al recibir

mensaje1 ▾

El bloque inicia la secuencia de comandos cuando se emite un mensaje con el nombre especificado seleccionado de la lista desplegable. El mensaje se define en el **bloque de emisión de mensajes**. La emisión y recepción de mensajes permite la comunicación entre sprites y con el escenario. De este modo, un sprite puede ejecutar el script de otro sprite.

Gato

al presionar

decir ¡Te atraparé ratón!

enviar te atraparé ▾

Ratón

al recibir te atraparé ▾

decir ¡Y yo correré muy lejos!



5.6

enviar

mensaje1 ▾

Este bloque nos permite definir y difundir un mensaje. Podemos elegir el nombre del mensaje en la lista desplegable o definir un nuevo mensaje utilizando la opción **Nuevo mensaje...** disponible al desplegar la lista.



Aparecerá entonces un cuadro de diálogo en el que introduciremos el nombre del nuevo mensaje.

Nuevo Mensaje✕

Nombre para el nuevo mensaje:

Cancelar

De acuerdo

Una vez emitido el mensaje (se ha ejecutado este bloque), se activan automáticamente los scripts que comienzan con los bloques **cuando recibo el nombre del mensaje**. El script que emite el mensaje continúa simultáneamente.

5.7 

Funciona de la misma manera que el **Enviar mensaje** (véase la descripción de ese comando). La diferencia es que el funcionamiento del script en el que se emitió el mensaje se pausa hasta que finaliza el funcionamiento de todos los scripts que reciben este mensaje.

6. Control

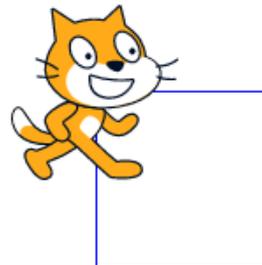
La categoría **Control** contiene bloques muy importantes, equivalentes a las instrucciones que controlan los lenguajes de programación (bucles, instrucciones condicionales). A continuación se describen todos los bloques de esta categoría.



Este bloque detiene la ejecución del código durante un tiempo especificado. El valor predeterminado se puede editar.



El bloque repetirá un conjunto de instrucciones un número determinado de veces. El número predeterminado de repeticiones puede editarse.



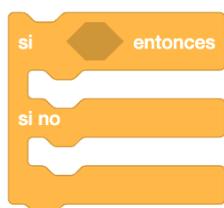
El bloque repetirá un conjunto de instrucciones para siempre, hasta que se pulse el botón de interrupción -el círculo rojo-. Si el script con el bloque **Por siempre** debe detenerse por sí mismo, debe utilizarse una instrucción condicional dentro de él y del bloque **Detener**.



6.4

Un bloque de instrucciones condicional; un conjunto de instrucciones se ejecutará una vez si una condición lógica insertada después de la palabra **Si** es verdadera. Las condiciones se construyen con el uso de bloques de las categorías **Sensores** y **Operadores**.





6.5

Un bloque de instrucciones condicional; un conjunto de instrucciones se ejecutará entre las palabras **entonces** y **si no** si una condición lógica insertada después de la palabra **si** es verdadera. En caso contrario, se ejecuta el conjunto de instrucciones que sigue a la palabra otherwise. Las condiciones se construyen con el uso de bloques de las categorías **Sensores** y **Operadores**.

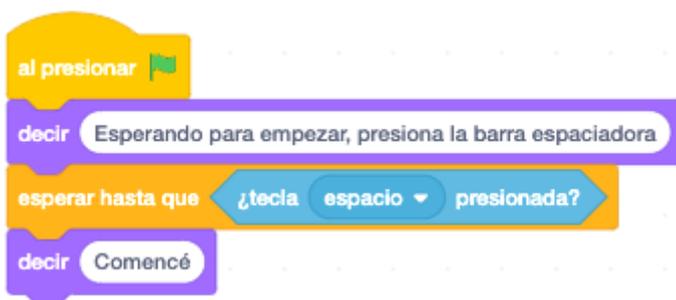


Estoy en el lado derecho de la pantalla



6.6

Suspende la ejecución del script hasta que se cumpla la condición lógica. Las condiciones se construyen con el uso de bloques de las categorías **Sensores** y **Operadores**.



Esperando para empezar, presiona la barra espaciadora





6.7

El bloque repetirá un conjunto de instrucciones hasta que se cumpla la condición lógica. Las condiciones se construyen con el uso de bloques de las categorías **Sensores** y **Operadores**.



6.8

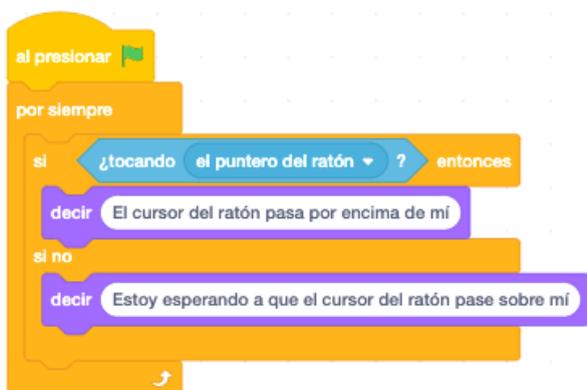
Detiene la ejecución de uno o varios scripts. En la lista desplegable podemos seleccionar: detener el script en el que se ha colocado un bloque (**este programa**), detener todos los scripts de todos los sprites y escenas (**todos**) o detener otros scripts del sprite a excepción del script en el que se ha iniciado este bloque (**otros programas en el objeto**).

7. Sensores

La categoría **Sensores** incluye bloques relacionados con el reconocimiento de diferentes situaciones que ocurren en el escenario relativas, entre otros, a los sprites y al ratón, bloques para extraer datos del teclado, y bloques asociados al temporizador, la fecha y la cámara. A continuación se describen la mayoría de los bloques de esta categoría, excepto los de funcionamiento de la cámara y fecha.

7.1

Comprueba si el sprite está tocando el elemento seleccionado en la lista desplegable. Puede ser cualquier otro sprite, el cursor del ratón o el borde del escenario. Este bloque se utiliza más a menudo en conjunción con uno de los siguientes bloques: **if**, **repetir hasta**, **esperar hasta**.



Estoy esperando a que el cursor del ratón pase sobre mí

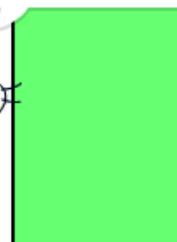


7.2

Comprueba si el sprite está tocando un color especificado en el escenario. Para seleccionar un color debemos hacer click con el ratón en la casilla con el color en el bloque, y luego seleccionar el color de entre los colores disponibles utilizando el cursor del ratón. Este bloque se utiliza más a menudo en conjunción con uno de los siguientes bloques: **if**, **repetir hasta**, **esperar hasta**.



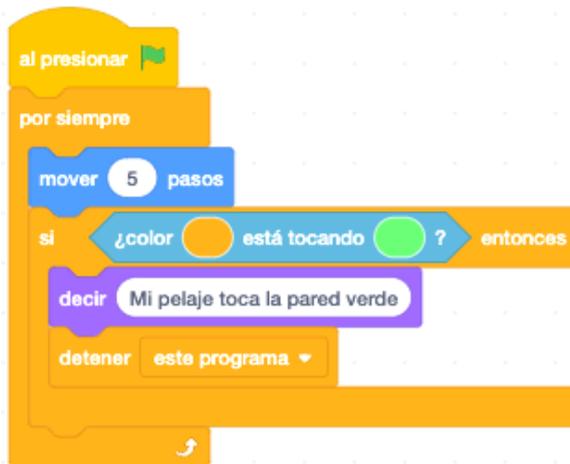
He alcanzado la muralla verde



7.3



Comprueba si el color del personaje sprite seleccionado está tocando un color especificado en el escenario. Para seleccionar un color debemos hacer clic con el ratón en la casilla con el color en el bloque, y luego seleccionar el color de entre los colores disponibles utilizando el cursor. Este bloque se utiliza más a menudo en conjunción con uno de los siguientes bloques: **if**, **repetir hasta**, **esperar hasta**.



7.4



La distancia actual del sprite desde un elemento seleccionado de la lista desplegable - cursor del ratón u otro sprite. Puede usarse como argumento en expresiones. La distancia se mide desde el centro del objeto.



7.5

preguntar ¿Cómo te llamas? y esperar

Muestra en una burbuja el texto introducido en el cuadro de bloque, así como una ventana con un cuadro de respuesta en el escenario. Espera a que se introduzcan los datos en el campo de edición y se confirmen pulsando Intro o haciendo click en el cuadro de edición. La información introducida mediante el teclado está disponible en guiones con el uso del bloque **respuesta**.

respuesta

¿Cuál es tu edad?



28

7.6

respuesta

La respuesta proporcionada por el usuario, como resultado del bloque **Preguntar y esperar**. Puede utilizarse como argumento en expresiones. Al marcar la casilla se muestra la respuesta proporcionada en el escenario.

7.7

¿tecla espacio presionada?

Comprueba si se ha pulsado la tecla seleccionada en la lista desplegable. Este bloque se utiliza más a menudo junto con uno de los siguientes bloques: **si, repetir hasta, esperar hasta**.

al presionar

decir Presiona la barra espaciadora

esperar hasta que ¿tecla espacio presionada?

decir La barra espaciadora ha sido presionada

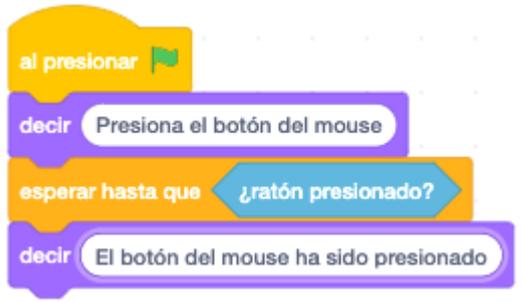
La barra espaciadora ha sido presionada



7.8

¿ratón presionado?

Comprueba si se ha pulsado el botón izquierdo del ratón. Este bloque se utiliza más a menudo junto con uno de los siguientes bloques: **if**, **repetir hasta**, **esperar hasta**.



El botón del mouse ha sido presionado



posición x del ratón

posición y del ratón

7.9

El valor actual de las coordenadas x e y del cursor del ratón. Pueden utilizarse como argumento en expresiones. Estas coordenadas también se muestran debajo de la esquina inferior derecha del escenario.

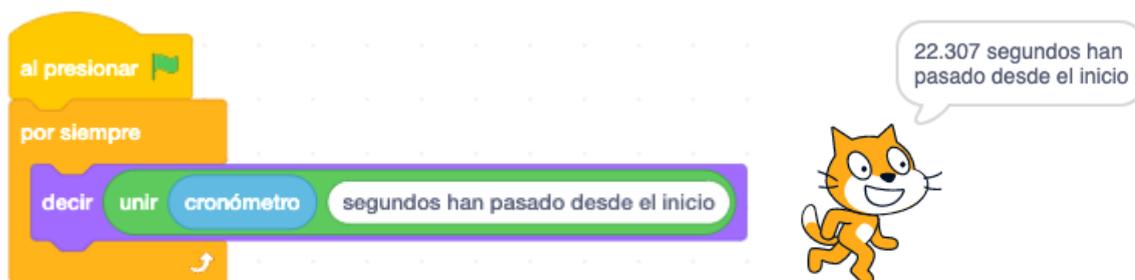


Estoy a la derecha del cursor del mouse



7.10 cronómetro

Tiempo actual en segundos desde el último click de la bandera verde o la ejecución del **reinicio cronómetro**. Se puede utilizar como argumento en expresiones. Al marcar la casilla se muestra el tiempo en el escenario.



7.11 reiniciar cronómetro

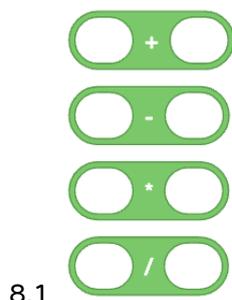
Reinicia la cuenta del temporizador. Los valores proporcionados por el bloque del temporizador volverán a contarse desde cero.

7.12 posición en x de Objeto1

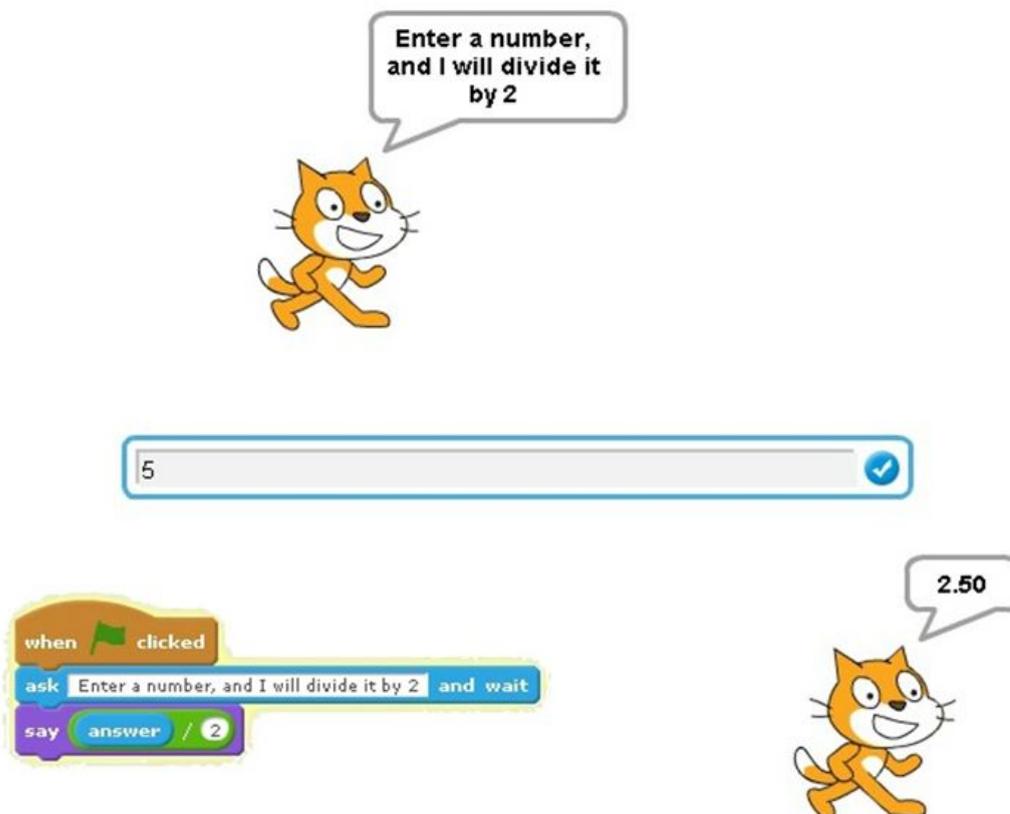
El valor actual seleccionado en la primera lista desplegable, perteneciente al sprite seleccionado en la segunda lista o al escenario. Podemos seleccionar la coordenada x o y, la dirección, el número del traje, el nombre o el tamaño. Para el escenario podemos elegir el número o el nombre del fondo. El valor devuelto se puede utilizar en expresiones.

8. Operadores

La categoría **Operadores** agrupa bloques de operaciones aritméticas básicas, operaciones lógicas y funciones diversas (tanto aritméticas como sobre textos). Todos los bloques de esta categoría se utilizan como argumentos para otros bloques. A continuación se describen la mayoría de los bloques de esta categoría.



Bloques de las cuatro operaciones aritméticas básicas (suma, resta, multiplicación, división). En los cuadros de edición deben proporcionarse argumentos (que pueden ser números introducidos manualmente, el valor de una variable o el resultado de otra acción, por lo que los bloques pueden incrustarse). Una expresión aritmética construida debe utilizarse como argumento de otro bloque, por ejemplo, para determinar el valor de una variable o de una expresión lógica.



Enter a number,
and I will divide it
by 2

when clicked
ask Enter a number, and I will divide it by 2 and wait
say answer / 2

2.50

8.2 elegir número al azar entre 1 y 10

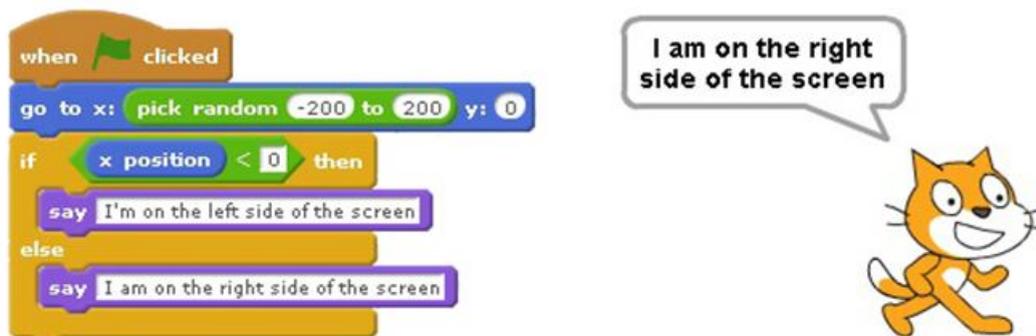
Extrae un número de un rango especificado. En el cuadro de edición debemos especificar el rango inferior y superior del número aleatorio a elegir. El bloque debe utilizarse como argumento para otro bloque.

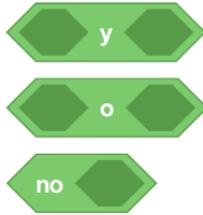


8.3

> 50
< 50
= 50

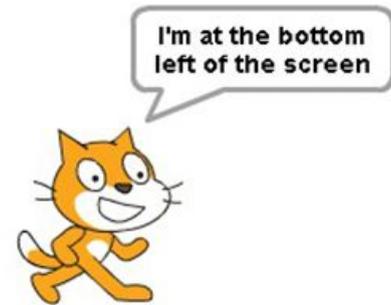
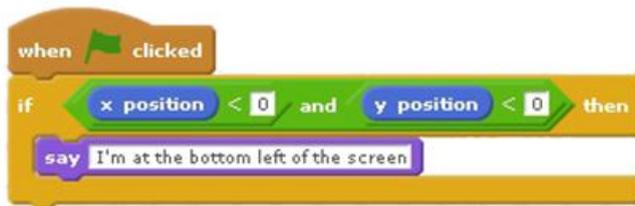
Bloques de comparación (menor que, igual a, mayor que). En los campos de edición debemos especificar los valores a comparar. Podemos introducir valores específicos en los campos de edición, la mayoría de las veces uno de los argumentos es el valor de una variable o un valor leído con el uso de otro bloque (por ejemplo, una coordenada de la posición de un sprite en el escenario). Este bloque se utiliza más a menudo en las siguientes instrucciones: **if**, **repeat until**, **wait until**.



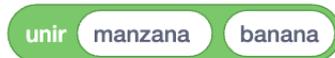


8.4

Los bloques utilizados para construir expresiones lógicas complejas (conjunción - ambas condiciones son verdaderas, alternativa - una de las condiciones es verdadera, negación - contradicción). Inserte los bloques coincidentes que devuelven valores booleanos (verdadero o falso) en cajas de argumentos. Los bloques se utilizan con mayor frecuencia en las siguientes instrucciones: **si**, **repetir hasta**, **esperar hasta**.



8.5



El bloque sirve para unir dos cadenas de texto en una. Normalmente, uno de los argumentos es el texto introducido y el otro un valor devuelto por otro bloque. El bloque se utiliza a menudo como argumento en los bloques **Decir** y **Pensar**.



8.6

módulo

El valor es el resto de la división del primer argumento por el segundo. El bloque se utiliza a menudo como argumento en operaciones aritméticas y lógicas.



8.7

redondear

El valor es el argumento redondeado al número entero más próximo - según las reglas de redondeo, es decir, 0,5 y más se redondea hacia arriba, en caso contrario hacia abajo. El bloque se utiliza a menudo como argumento en otros bloques.

8.8

valor absoluto de

El valor es el resultado de la función seleccionada en la lista desplegable para el argumento especificado en el cuadro de edición. En la lista desplegable podemos seleccionar las siguientes funciones (y más):

- valor absoluto
- piso
- techo
- raíz cuadrada
- sen
- cos
- tan



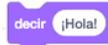
Actividades en Scratch

1. Nuestro primer script (programa): "Hola"

1.1 ¡Es el momento de entrar en acción! Sigue cada una de las siguientes instrucciones en el programa Scratch

Elige el menú "Apariencia"

Agregue el bloque "decir ¡Hola!"



Elige el menú "Eventos"

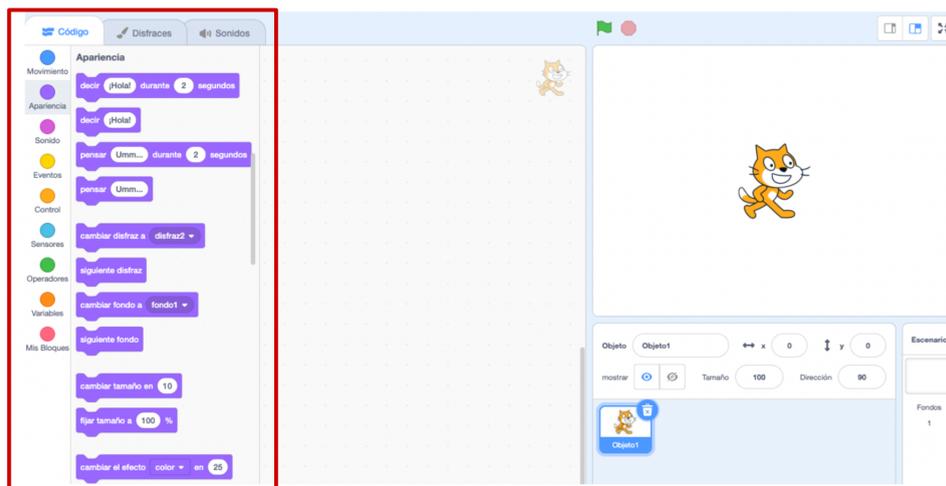
Agregue el bloque "al presionar"



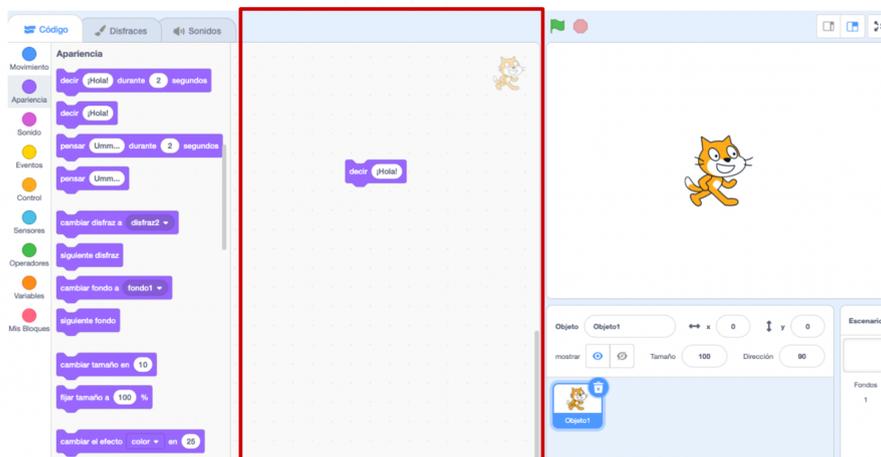
Comprueba el comportamiento de tu sprite a la derecha presionando 🚩

1.2 ¿Tienes dudas? Aquí va la solución

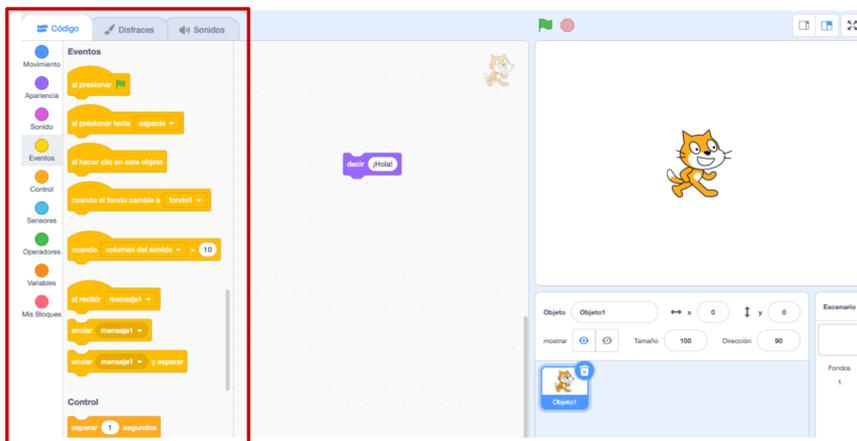
Elige el menú "Apariencia"



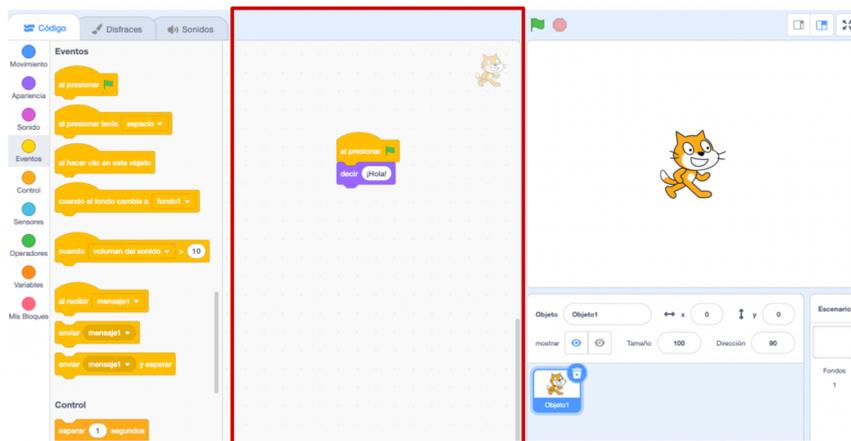
Agregue el bloque "decir ¡Hola!"



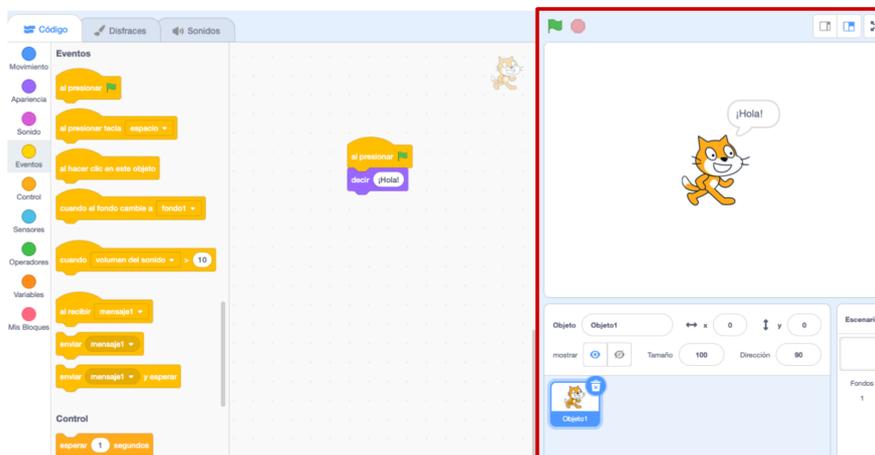
Elige el menú “Eventos”



Agregue el bloque “al presionar” 



Comprueba el comportamiento de tu sprite a la derecha presionando 



¡Enfrentemos el siguiente desafío!

Implemente un programa donde el sprit sólo diga
“¡Hola a todos!” por 10 segundos

2. Probemos ahora otros elementos del menú “Apariencias”

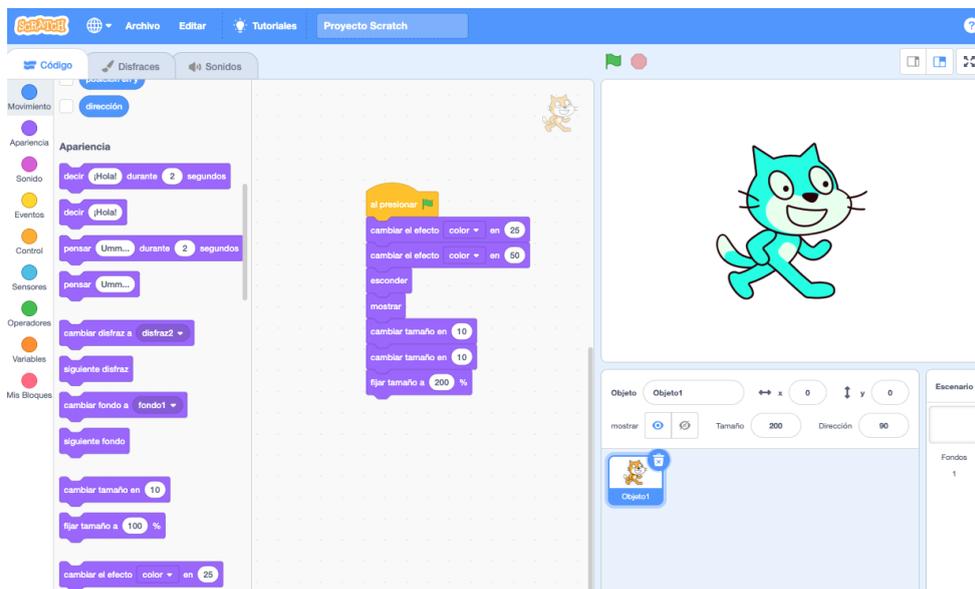
2.1 Pruebe individualmente cada una de los siguientes elementos

- Cambiar el efecto de color en 25 
- Esconder 
- Mostrar 
- Cambiar tamaño en 10 
- Fijar tamaño a 100% 

2.2 Nuevo desafío: implemente la siguiente secuencia de instrucciones

- Cambiar el efecto de color en 25
- Volver a cambiar el efecto de color, ahora en 50
- Esconder
- Mostrar
- Cambiar tamaño en 10
- Volver a cambiar el tamaño en 10
- Fijar tamaño a 200%

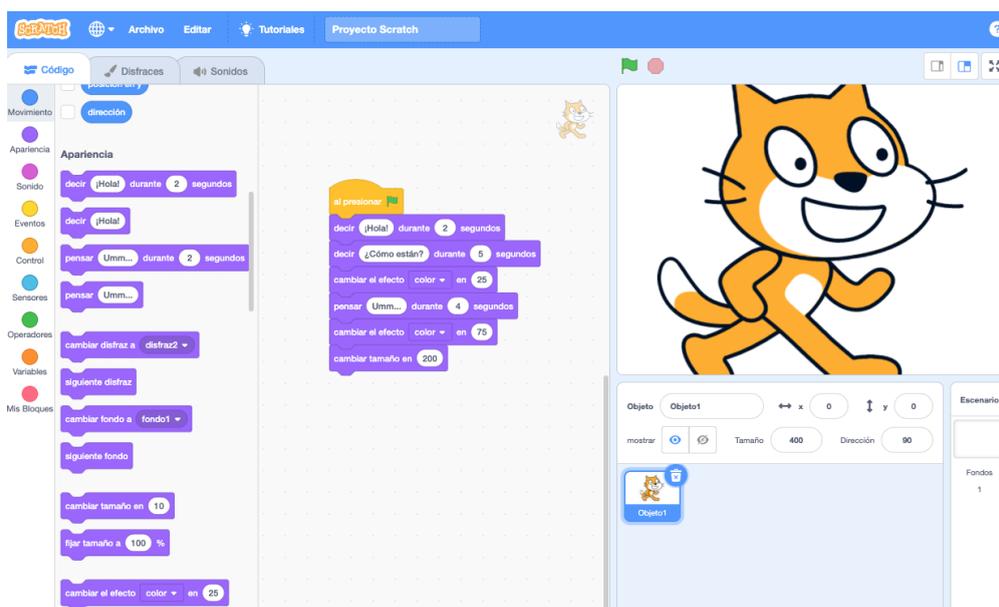
2.3 Solución



2.4 Nuevo desafío: implemente la siguiente secuencia de instrucciones

- Diga "Hola" durante 2 segundos
- Pregunte "¿Cómo están?" durante 5 segundos
- Cambie el color en un 25
- Luego piense "Hmm.." durante 4 segundos
- Cambie el color un 75
- Cambie el tamaño en 200

2.5 Solución

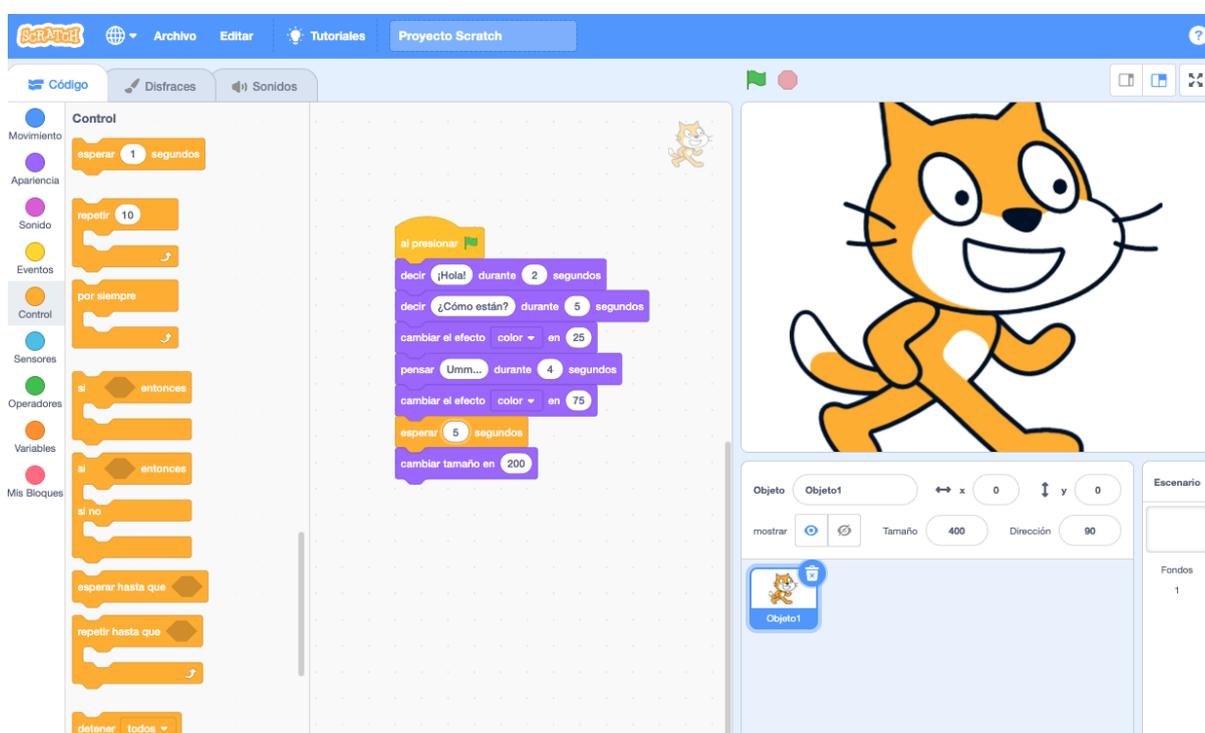


3. La instrucción “esperar”

La instrucción **ESPERAR** es necesaria para relentizar la actuación de nuestro personaje, de forma que podamos ver u oír cada instrucción correctamente (los computadores son demasiado rápidos a veces)

Para obtener la instrucción **ESPERAR**, dirijase al menú CONTROL. Ignore las otras opciones del menú por el momento.

Ahora, Inserte un bloque **ESPERAR** en nuestro script de apariencias previamente implementado.

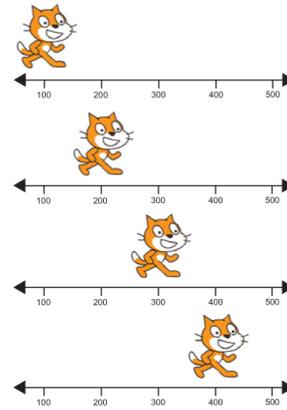
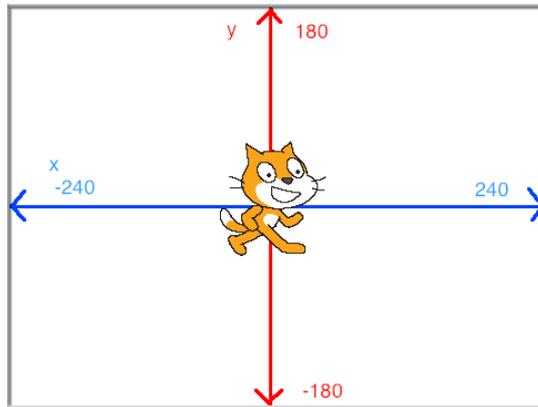


¡Enfrentemos el siguiente desafío!

Debe implementar un relato interactivo entre dos personajes (a elección), utilizando los ítems del menú Apariencias, Eventos y Control.

4. Aprendiendo a movernos

El menú de Movimientos es el encargado de (sorpresa) mover a nuestro personaje. Todos los movimientos que necesites los encontrarás aquí (a menos que no exista).



4.1 ¡Aprendamos a posicionar a nuestro sprite en el escenario!

Elige el menú “Apariencia”

Agregue el bloque “fijar tamaño a %”

fijar tamaño a 100 %

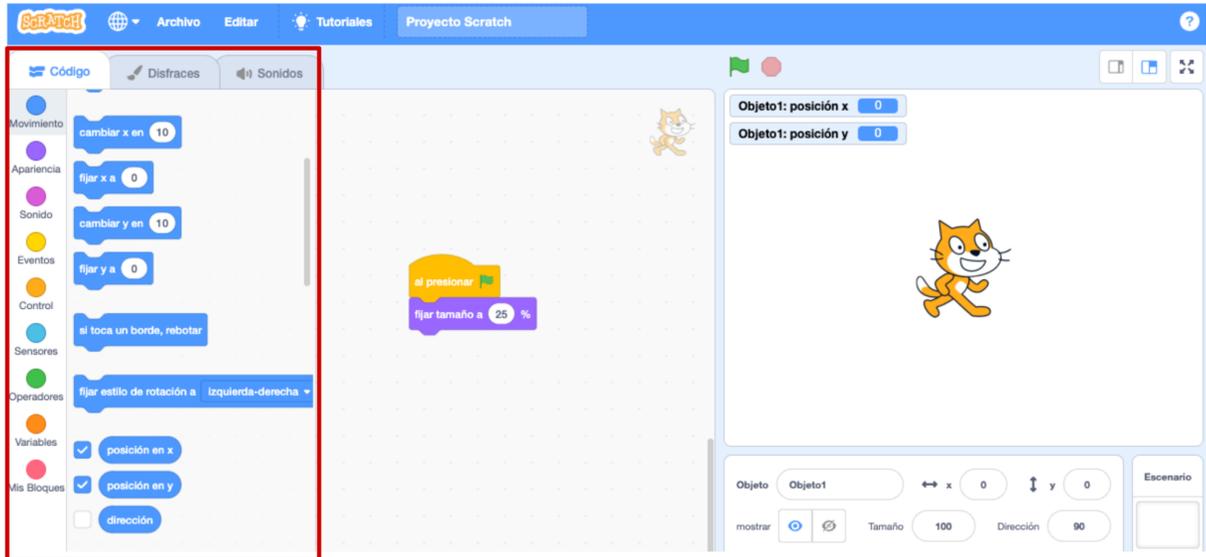
Cambie el valor de 100% a 25% en el bloque anterior

Elige el menú “Eventos”

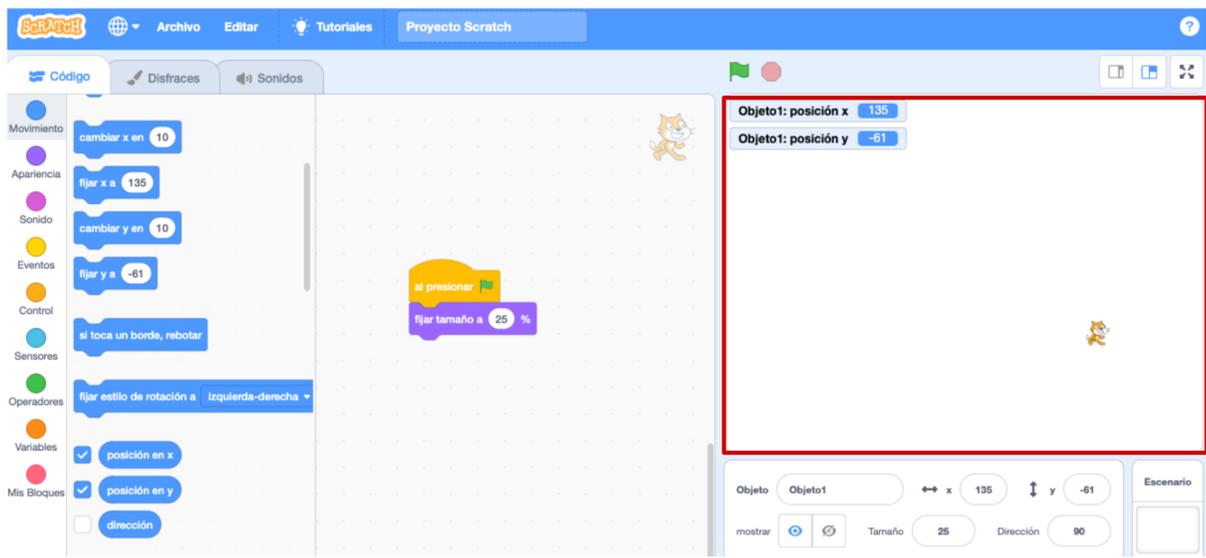
Agregue el bloque “al presionar”

al presionar

- Elige el menú “Movimiento”
- Marque las casillas: “posición en x”, “posición en y”



- Dirija su mouse al sprite y arrástrelo en el escenario (usando el mouse de su computador). Observe cómo cambian los valores de la posición x-y en la esquina superior izquierda.



4.2 ¡Aprendamos a ir a ciertas locaciones en el escenario!

Elige el menú “Movimiento”

Agregue el bloque “ir a x: y:”  en el panel del script

Edite las coordenadas entre los siguientes valores:

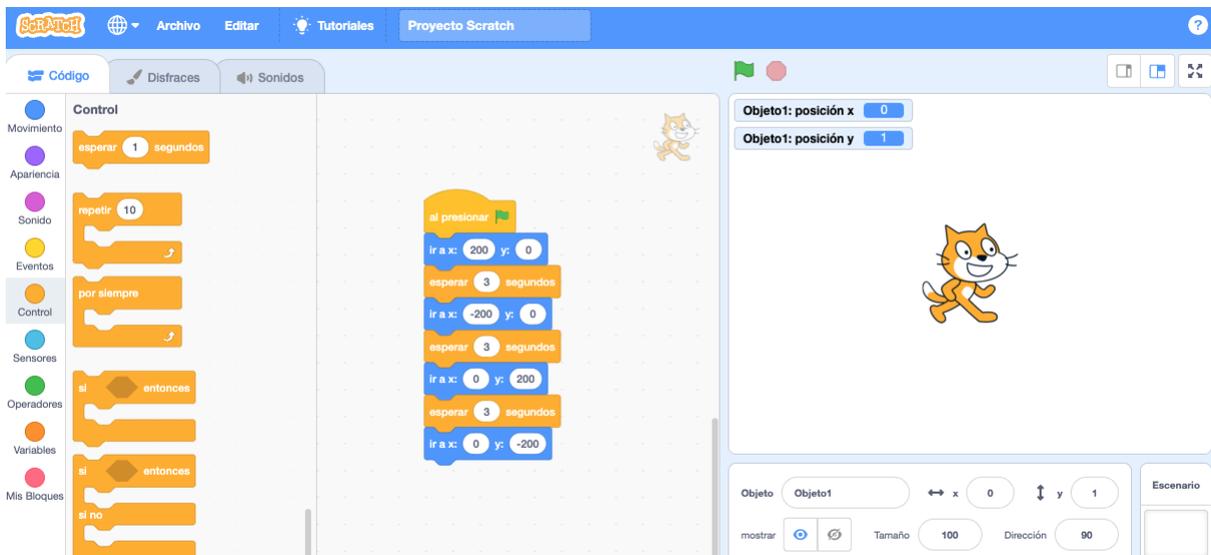
- x=200, y=0
- x=-200, y=0
- x=0, y=200
- x=0, y=-200

Elige el menú “Control”

Emplee el bloque “esperar segundos”  entre cada instrucción (defina la cantidad de segundos)

Elige el menú “Eventos”

Agregue el bloque “al presionar” 

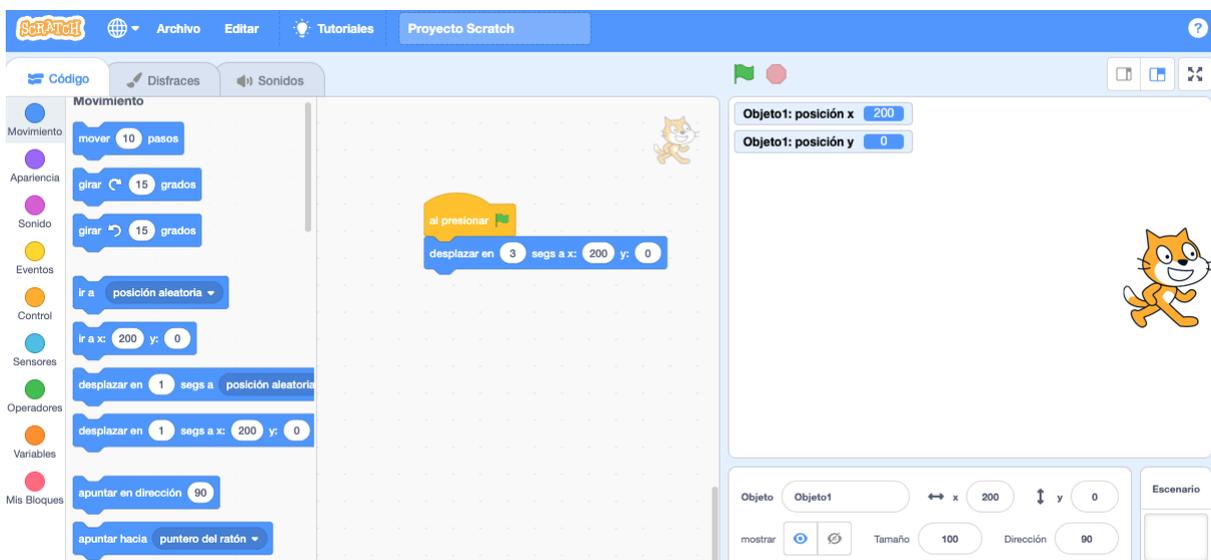


En base al código implementado...

¿Hacia dónde va el personaje?

4.3 ¡Aprendamos a desplazarnos en el escenario!

- Elige el menú “Movimiento”
- Agregue el bloque “desplazar en segundos a x: y:”  en el panel del script
- Edite los valores de las coordenadas y el tiempo de desplazamiento
- Elige el menú “Eventos”
- Agregue el bloque “al presionar 



En base al código implementado...

¿Hacia dónde va el personaje?
¿Y cuál es su comportamiento?

4.4 Nuevo desafío: ¡Recorramos un triángulo!

Cree un script para deslizar el personaje a lo largo de los lados de un triángulo.

- El primer vértice del triángulo es (-100, -100).
- El segundo vértice es (200, -100).
- El tercer vértice es (50, 100).

¡Asegúrese de completar todo el triángulo!

4.5 Nuevo desafío: ¡Démosle vida a nuestro personaje!

Implemente el siguiente script para dar vida a Scratch, incluyendo bloques de espera:

- Empezar en (-100, 100)
- Mover 200 pasos
- Girar 90 grados a la derecha
- Mover 200 pasos
- Girar 90 grados a la derecha
- Mover 200 pasos
- Girar 90 grados a la derecha
- Mover 200 pasos

¡Utilice los bloques de espera para que el movimiento parezca suave!

4.5 Gran desafío: ¡Vivamos la pasión del baseball!

Hoy es un gran día para el baseball en Chile: el estadio está lleno y todos esperan ansiosamente al primer bateo. Su personaje está nervioso pero con plena confianza de que será capaz de patear adecuadamente la primera bola.

Su misión es preparar el escenario para la transmisión. Para esto debe recrear un escenario similar al siguiente (seleccionando el sprite y el fondo respectivo):



Ahora, implemente el siguiente script para dar vida a su personaje, incluyendo bloques de espera:

- El bateador comienza en la base
- El bateador dice "Golpeo la bola" durante 2 segundos
- El bateador se mueve hacia la 1ª base
- El bateador cambia de color
- El bateador corre hasta la 2ª base
- El bateador dobla su tamaño
- El bateador dice "Hice un doble" durante 5 segundos
- El bateador corre hasta llegar a la base original

¡Le deseamos todo el éxito!

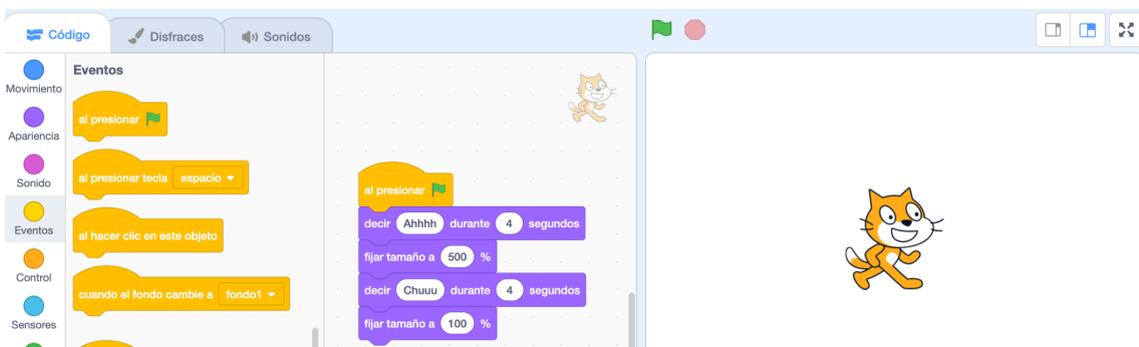
5. Aprendiendo a programar con repetición

5.1 El script del gato que estornuda

Implemente el siguiente script para ayudar a Scratch a estornudar, incluyendo bloques de espera:

- El gato debe decir "Ahhh" durante un rato
- Luego, el gato debe crecer 5 veces más grande
- Luego, el gato debe decir "Chuuu" durante un rato
- Luego, el gato debe encogerse 5 veces más pequeño

Solución:

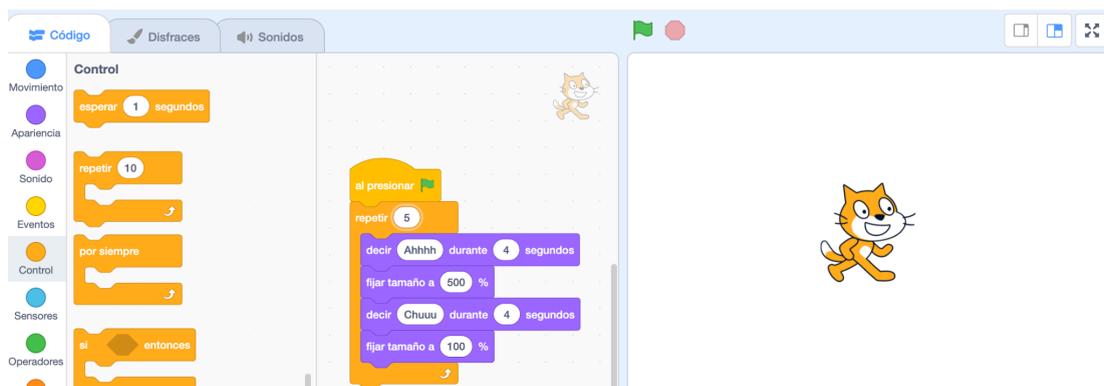


5.2 El script del gato que estornuda 5 veces

Implemente el siguiente script para ayudar a Scratch a estornudar 5 veces, incluyendo bloques de espera:

- El gato debe decir "Ahhh" durante un rato
- Luego, el gato debe crecer 5 veces más grande
- Luego, el gato debe decir "Chuuu" durante un rato
- Luego, el gato debe encogerse 5 veces más pequeño

Solución:



5.3 Desafío: El script del gato que hace ejercicio

Implemente un programa que permita al gato dar una vuelta sobre si mismo.

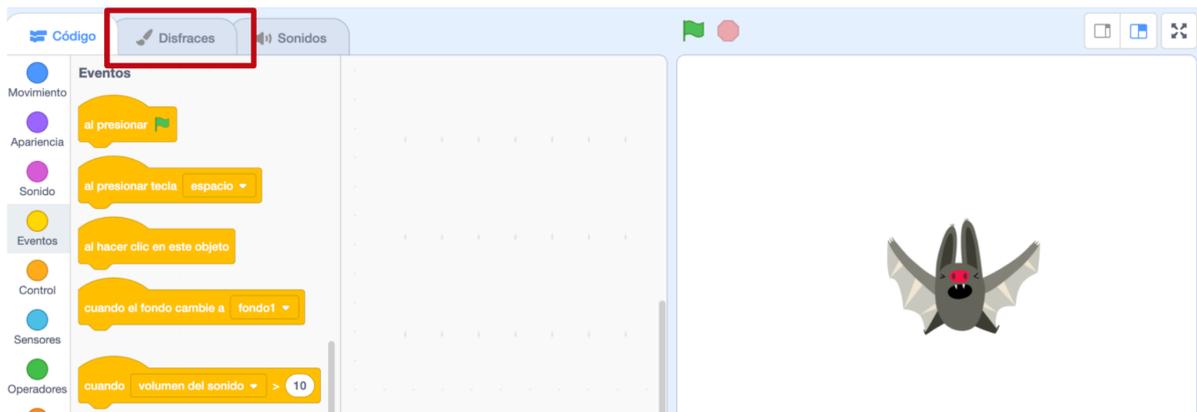


Hint: Para dar un giro completo, el sprite debe girar 20 grados, 18 veces por cada giro.

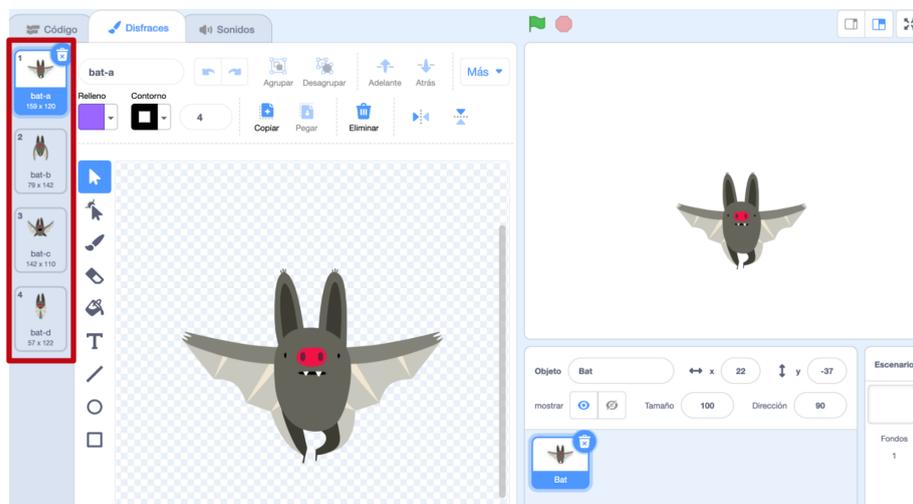
5.4 ¿Cómo hacer que un murciélago aletee?



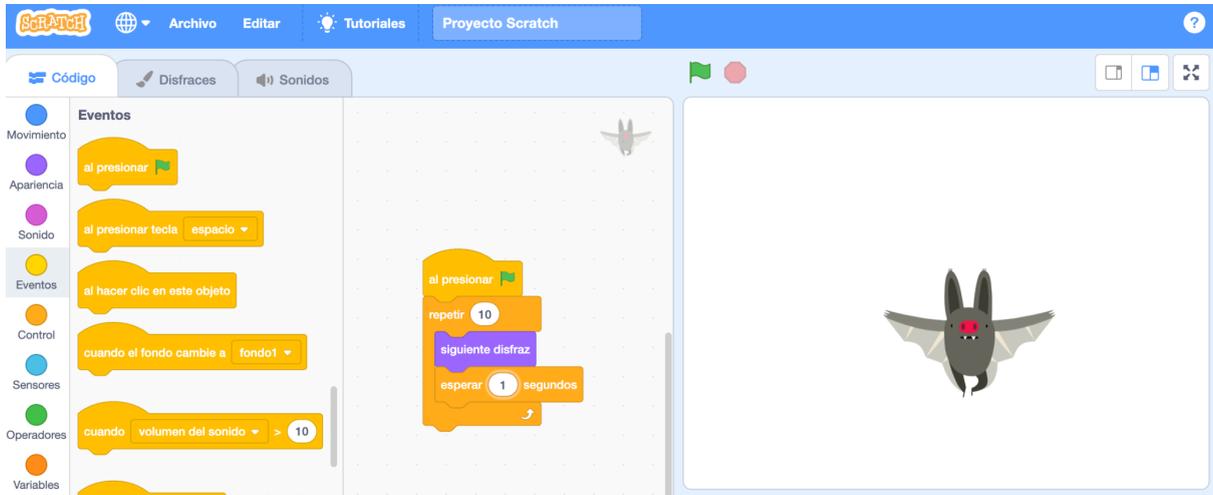
Elige la pestaña “Disfraces”



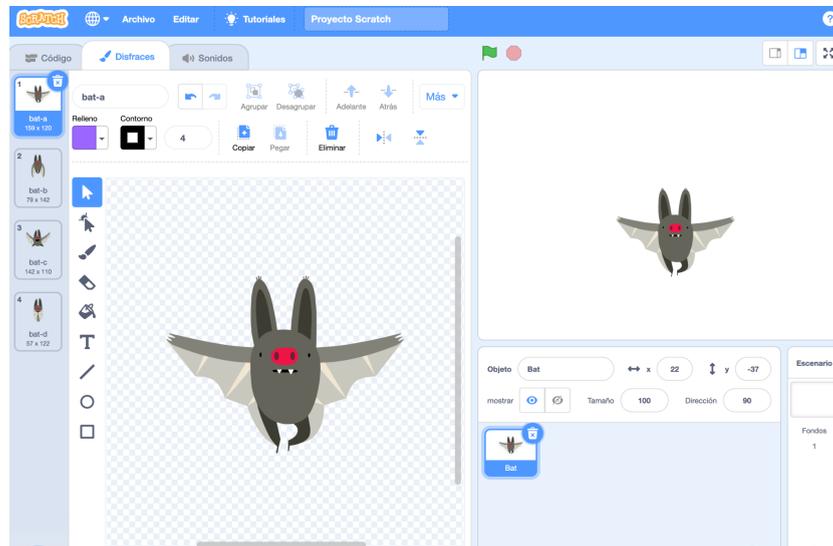
Observe los diferentes disfraces en el menú lateral de la izquierda

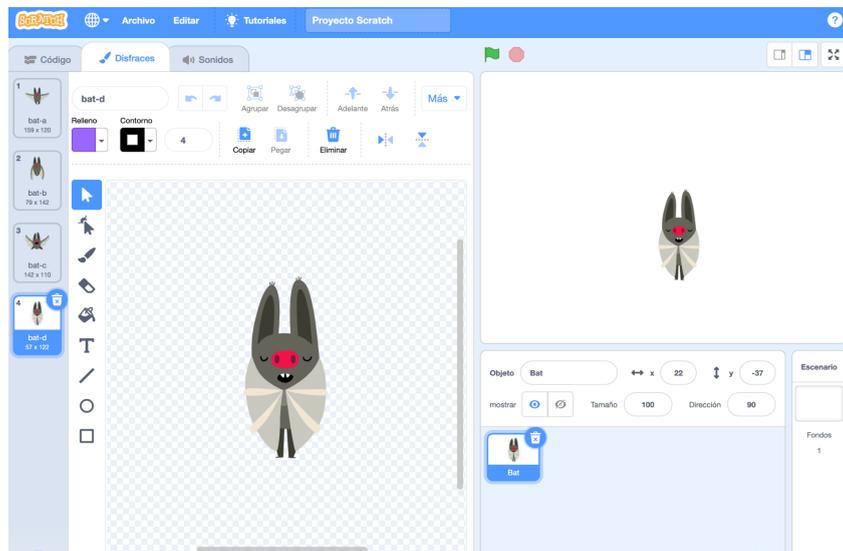
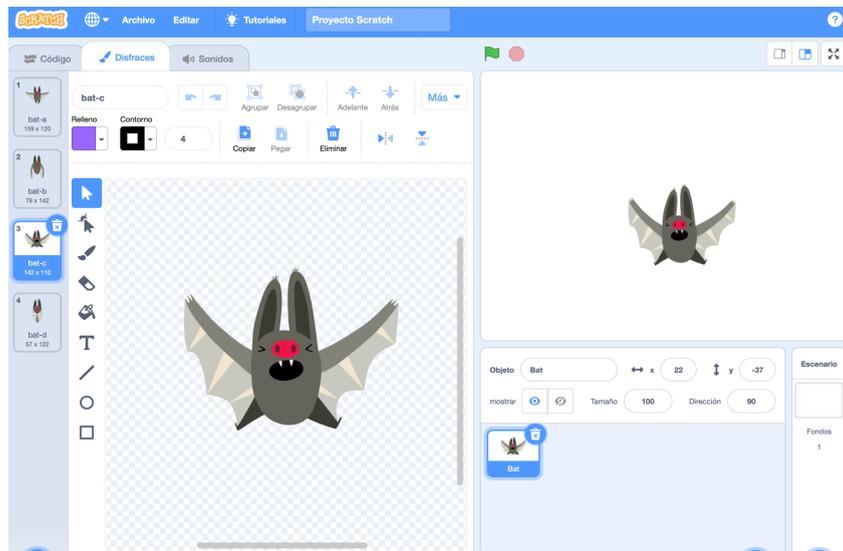
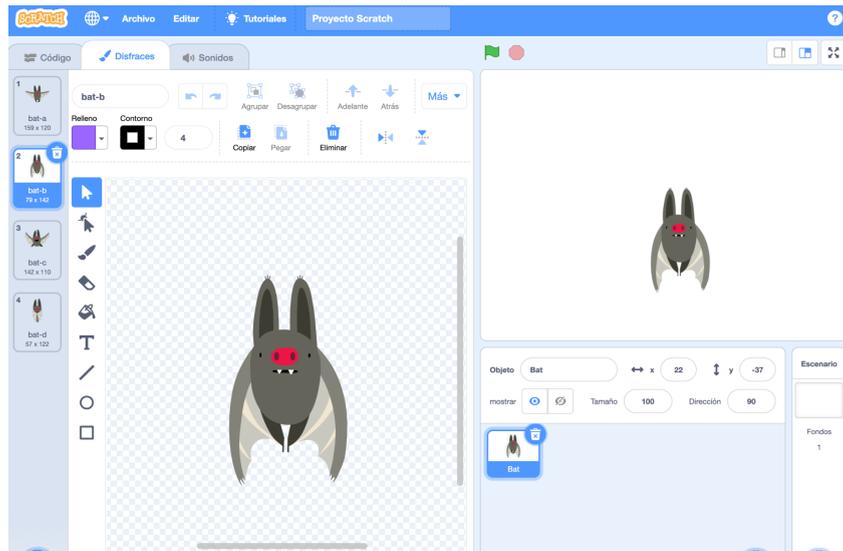


- Ahora, vuelva a la pestaña “Código” e implemente el siguiente script usando bloques del Menú “Eventos”, “Control” y “Apariencia”



- ¿Qué sucede cuando ejecutamos el programa anterior?
- Si vuelve a la Pestaña “Disfraces”, se dará cuenta que el script anterior cambia la apariencia del sprite secuencialmente en base a los disfraces que tiene asociado el personaje





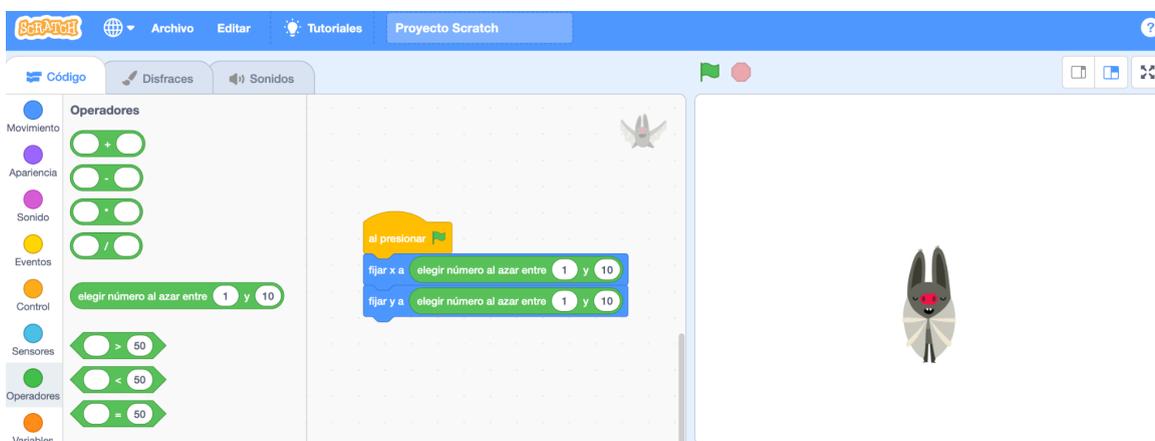
5.5 Desafío: El murciélago difícil de atrapar

Después de la pandemia del COVID-19, no quiere saber más sobre murciélagos. Sin embargo, para enseñar sobre este ser vivo a sus estudiantes, debe implementar un programa interactivo que muestre cómo vuelan los murciélagos y lo difícil que es atraparlos.

Para esto, usted debe implementar el siguiente script (incluyendo bloques de espera):

- Cambiar el disfraz del murciélago
- Establecer la posición X en un número aleatorio
- Establecer la posición Y en un número aleatorio
- Mover el murciélago a la posición (X,Y)
- Repetir la secuencia anterior una cantidad suficiente de veces, tal que se vea el movimiento del murciélago
- Por supuesto, ¡el murciélago debe permanecer en el escenario!

Solución



5.6 Desafío: Aprendiendo a controlar el movimiento del murciélago

Para la siguiente implementación, explore y use los diferentes bloques del Menú “Eventos”

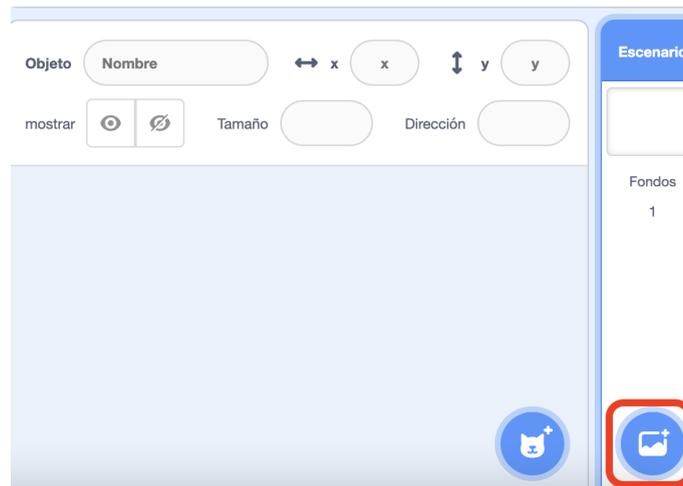
- Al presionar la barra espaciadora, el murciélago debe realizar movimientos aleatorios
- Al presionar la tecla derecha, el murciélago debe moverse a la derecha
- Al presionar la tecla izquierda, el murciélago debe moverse a la izquierda

El murciélago se comportará de forma diferente dependiendo de la tecla que se teclee.

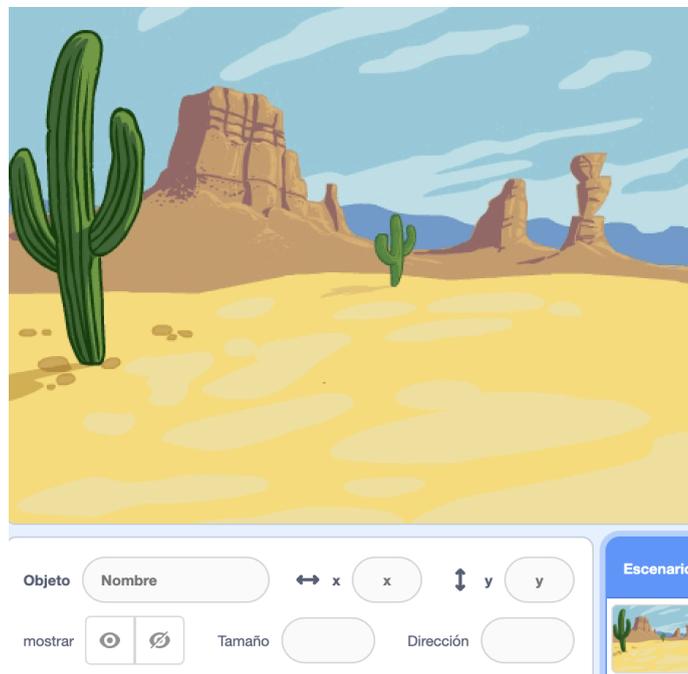
Cacería de Ratones

2.1 Seleccionando nuestros personajes y el escenario

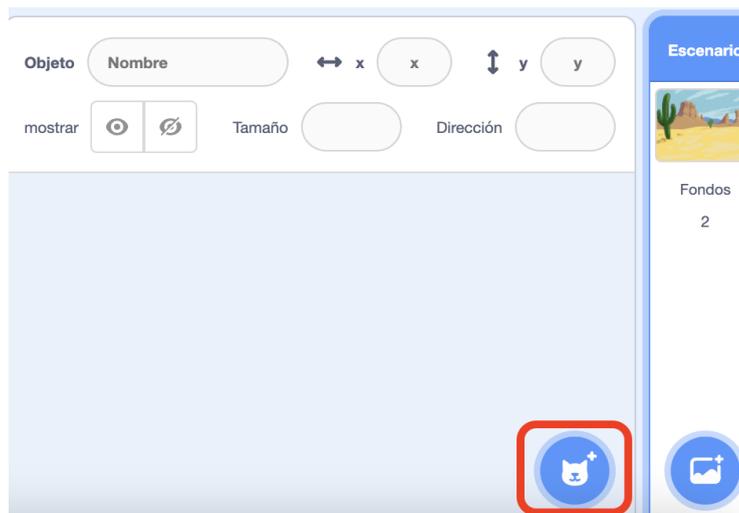
- a) En primer lugar, selecciona el escenario ideal para la actividad. En nuestro caso, seleccionaremos un desierto, tal como se aprecia en la siguiente secuencia.



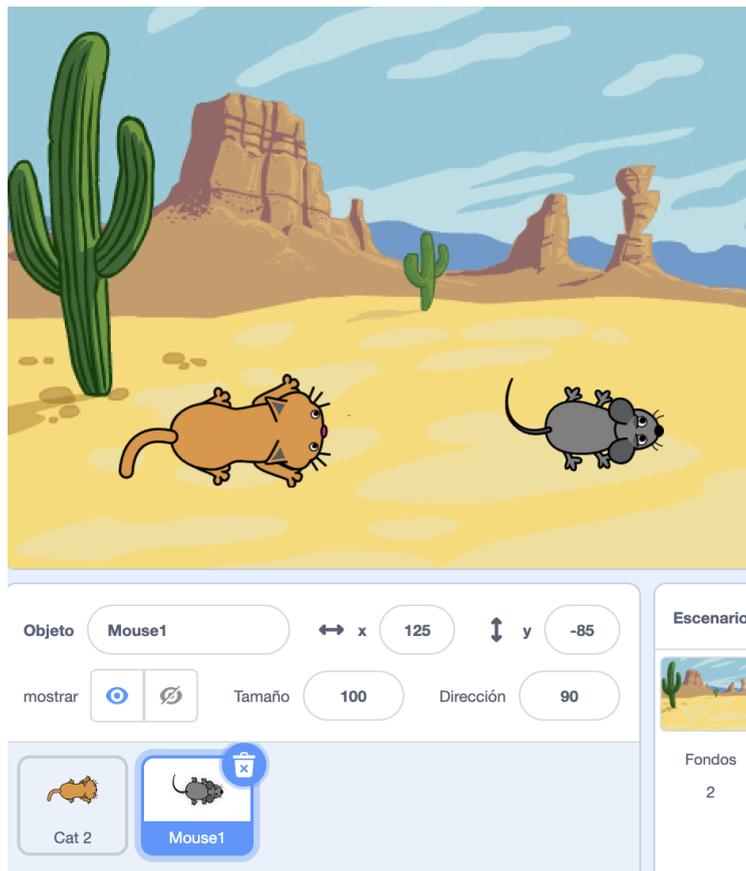
- b) La selección del escenario debería quedar similar a la siguiente imagen.



- c) Ahora, seleccionemos nuestros personajes. Para esto, escoge un visualización del gato visto desde arriba y un ratón, tal como se ve a continuación.



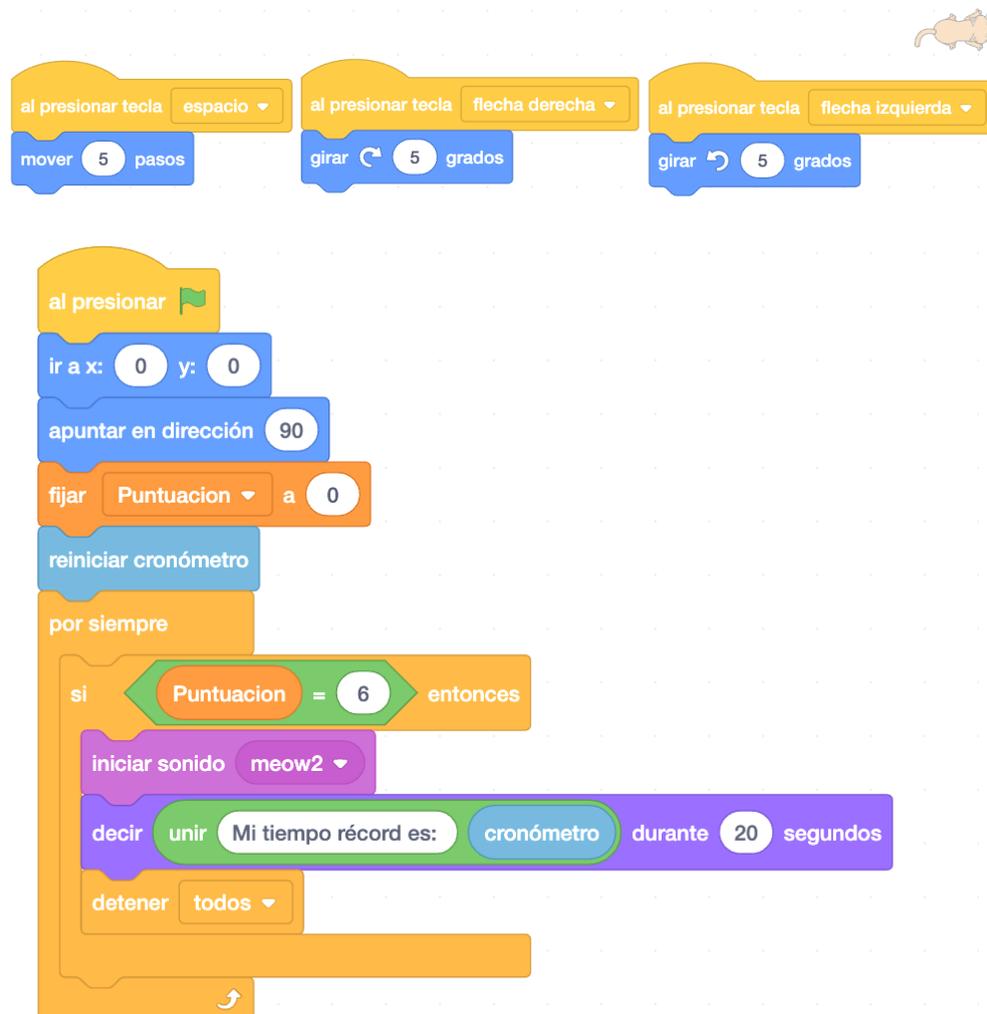
d) La selección de los personajes debería quedar similar a la siguiente imagen.



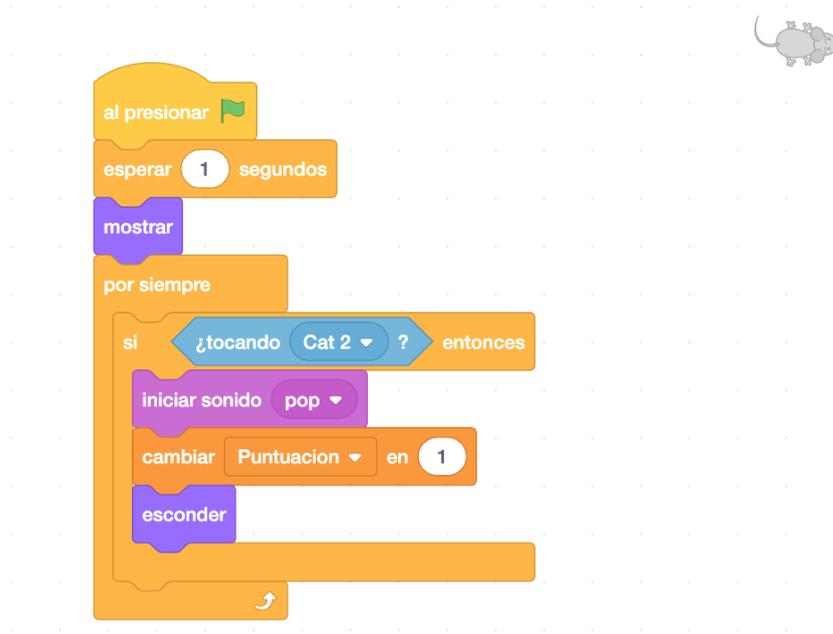
2.2 Programando la lógica del gato

Ahora, ¡implementemos a nuestro felino cazador! En primer lugar, deseamos definir los movimientos de nuestro personaje. Si presionamos a la derecha, el gato se mueve a la derecha.

Si presionamos a la izquierda, el gato se mueve a la izquierda. Si presionamos espacio, el gato se mueve hacia adelante.



2.3 Programando la lógica del ratón



```
al presionar [bandera verde]
esperar 1 segundos
mostrar
por siempre
  si [¿tocando Cat 2?] entonces
    iniciar sonido [pop]
    cambiar [Puntuacion] en 1
    esconder
```

2.4 Replicando nuestro ratón: Escenario final

