Original file

Manual Introductorio a Scratch

Scratch es un software de programación libre, que nos permite adentrarnos a esos primeros pasos para el aprendizaje de la programación. Scratch destaca porque su interfaz limpia e intuitiva, permite el acoplamiento de bloques entre si para construir pequeños programas, animaciones, videojuegos, y finalmente simples (o muy complejos) flujogramas.

Puedes instalar el software descargandolo desde aqui: https://scratch.mit.edu/scratch_1.4/+

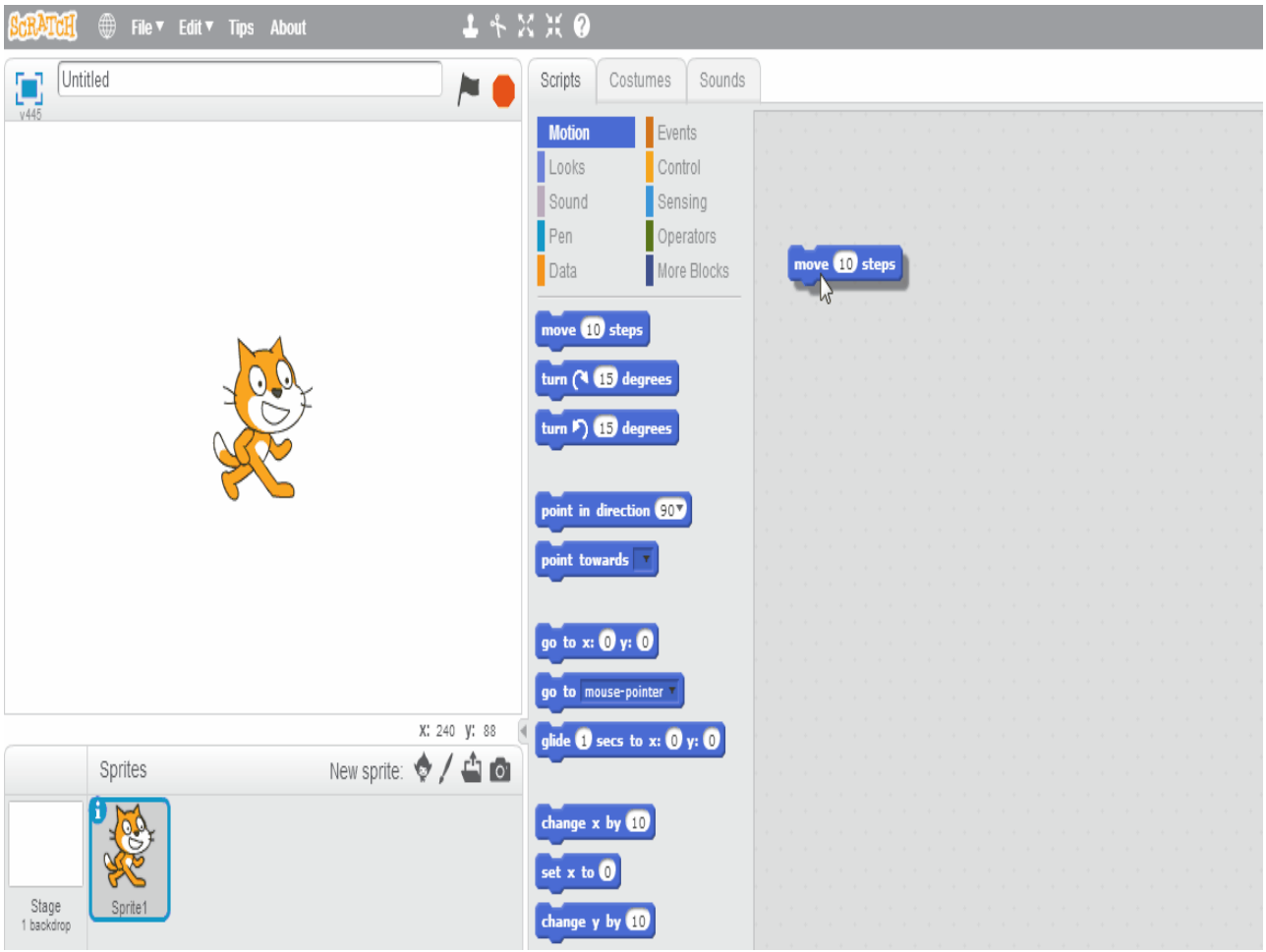
La mejor opción es probarlo en línea aqui: https://scratch.mit.edu/projects/editor/?tip_bar=getStarted

Conociendo la interfaz

Básicamente la interfaz de scratch se divide en 3 secciones importantes. A la izquierda, en la parte superior, la **paleta de bloques** es la sección donde se agrupan los diferentes bloques y sus respectivos usos seleccionados por color. En la parte inferior podemos acceder a cada bloque (funcionalidad) que pertenece a cada familia de color.

La parte central, es un lienzo donde se agrupan los diferentes bloques de la izquierda, arrastrándose

a esta parte central; para agrupar y crear los diferentes programas, que por lo general, son visualizados en el **escenario**



Para finalizar el tema de la interfaz, en esta parte central tenemos tres pestañas. En estas pestañas **Programas** será la parte donde crearemos el programa.**Disfraces** nos permite dibujar o importar diferentes imágenes de nuestro personaje. Scratch cuenta con una biblioteca llena de imágenes que podemos usar.

Por otro lado, en la parte superior derecha de la interfaz, la bandera verde de la parte superior sirve para ejecutar el programa, y el botón rojo para detener el programa.

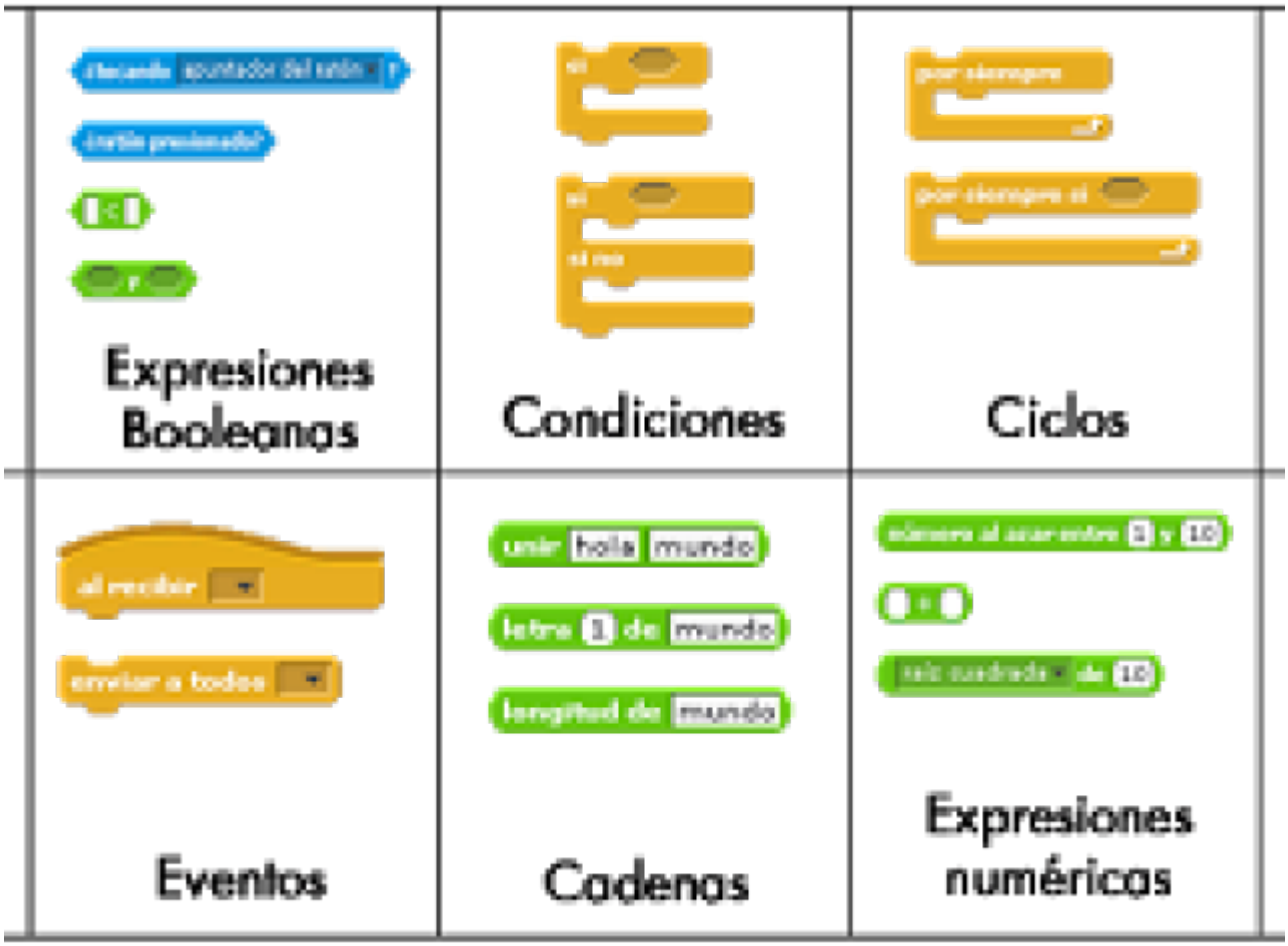
La paleta de bloques más detalladamente

Paleta de programación en la web

Paleta de colores instalado localmente

Las funcionalidades que presentan la interfaz web de la interfaz que se instala en escritorio difieren un poco, pero básicamente permiten hacer las mismas cosas salvo que en el aplicativo web; podemos crear nuestros propios bloques.

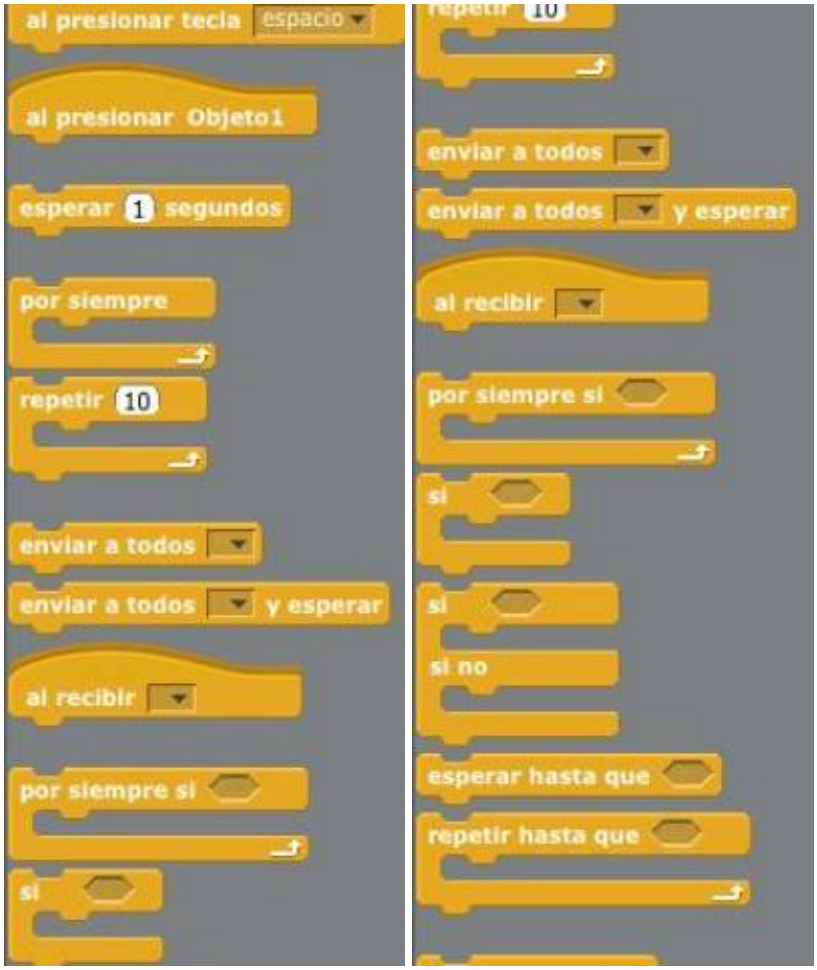
Como en cualquier lenguaje de programación, scratch no difiere de ellos en cuanto a su expresión algorítmica, pues cuenta de sentencias, condicionales, podemos crear variables y listas. Estas funcionalidades + otras que agrega scratch hace que se convierta en una gran opción para entender cómo funciona la lógica de la programación.



Los módulos de programación.

En esta sección encontraremos módulos que nos permitirán tomar decisiones dentro del programa, estos módulos permiten que nuestro programa tomen un camino u otro dependiente de lo que ocurra en su interior.

Aquí algunas opciones



¿ Que hace cada Bloque de control ?

“Al presionar bandera verde” : Este bloque con una curva es su parte superior, se usa siempre para dar inicio a un programa. Normalmente se ubica en la parte superior de un script.

“Al presionar la tecla ”espacio ” : Este bloque es igual que el anterior y tiene la misma funcionalidad, su único cambio es que nos ofrece la posibilidad de iniciar un programa usando una tecla del computador, para elegirla, hacemos clic donde dice “espacio” y saldrá un recuadro emergente que nos dará posibilidades de elección de teclas.

“Al presionar objeto1” : Este bloque es igual que el anterior y tiene la misma funcionalidad, cambia en que nos ofrece la posibilidad de iniciar un programa presionando un objeto en la pantalla.

“esperar 1 segundos” : En algunas ocasiones necesitamos que nuestro programa espere un tiempo determinado. Este bloque hace que nuestro script espere, ese tiempo puede ser 1 segundos por defecto. Si queremos otro tiempo, solo basta con hacer clic en el espacio del número y modificarlo a nuestro gusto.

“por siempre”: Este bloque ejecutará por siempre todo lo que esté adentro de este, este efecto es llamado “loop” porque es un acontecimiento que no tiene fin.

“repetir 10”: Este bloque funciona igual que el bloque anterior, su única diferencia es que es un loop que tiene fin, y ese fin está determinado por el número que escribamos, por defecto es 10.

“Enviar a todos”: Este bloque es un emisor para otros bloques. Se usa normalmente cuando

queremos que un acontecimiento se comunique con muchos bloques. Lo usaremos más adelante.

“Enviar a todos y esperar”: Este bloque hace lo mismo que el anterior, su diferencia es que cuando envía la información espera.

“Al recibir ”: Cuando se envía un mensaje con los dos bloques anteriores, este bloque los recibe y ejecuta lo que haya después. Efectivamente cuando se usa un bloque “enviar a todos”, se debe usar un bloque “al recibir”.

“por siempre si”: Este bloque crea un loop infinito de lo que hay en su interior siempre y cuando se cumpla la condición requerida en su icono de diamante.

“si”: Este bloque realiza todo lo que haya en su interior una vez si se cumple la condición requerida en su icono de diamante.

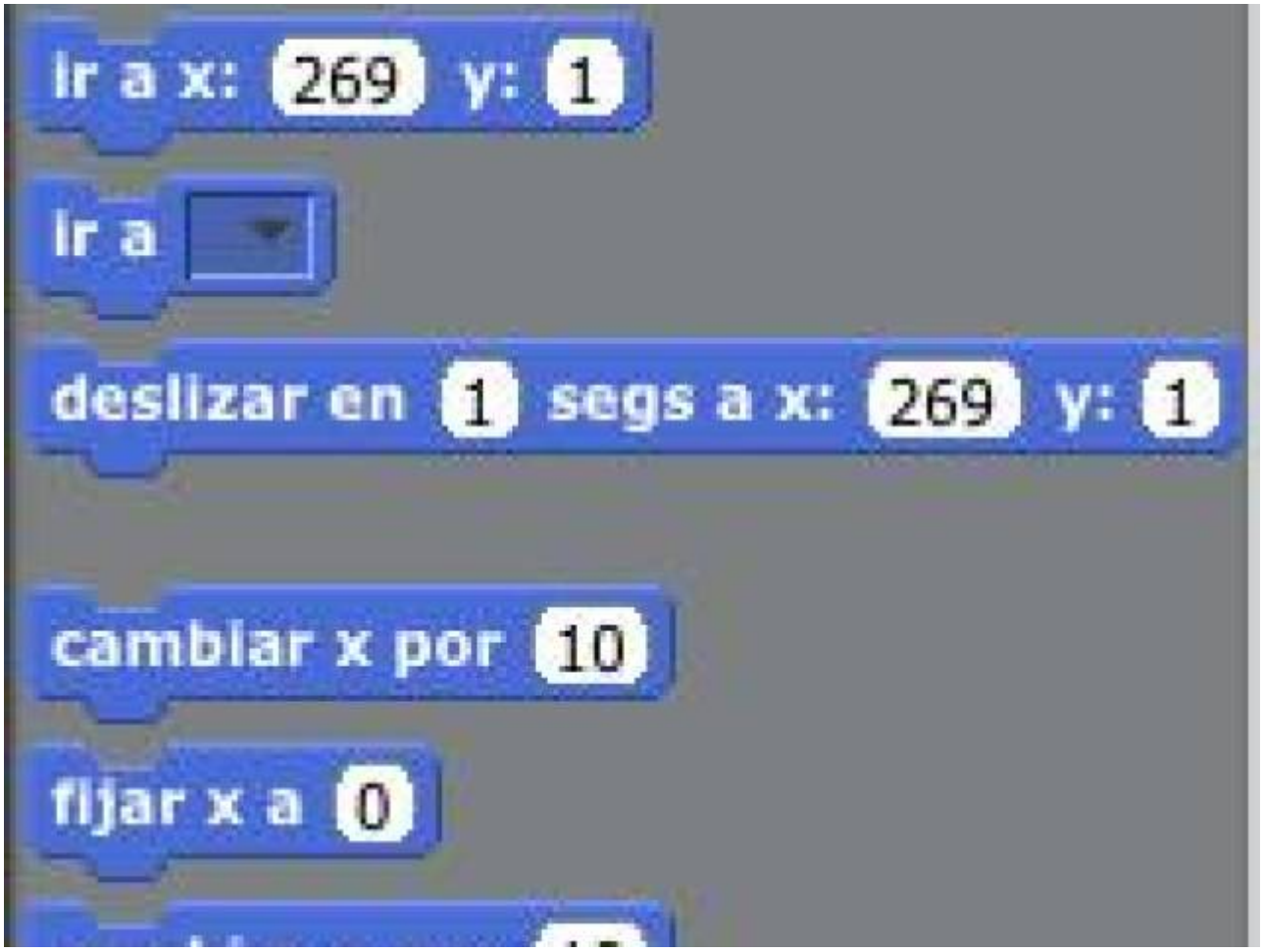
“si, si no”: Este bloque funciona exactamente igual que el bloque anterior, se diferencia en que si no se cumple la primera condición, el programa ejecutara lo que este adentro de “si no”, como alternativa a la primera opción.

“esperar hasta que”: Este bloque espera que suceda una condición para ejecutar una vez su contenido.

“repetir hasta que”: Este bloque ejecuta su contenido en forma de loop hasta que ocurra su condición.

Sección de Movimiento.

En esta sección encontramos bloques que nos permitirán desplazar y girar objetos a través del escenario.



¿ Que hace cada Bloque de control de movimiento?

“mover 10 pasos”: Este bloque hace que el objeto en el escenario se desplace en la zona del escenario. Si queremos que se desplace hacia la izquierda tenemos que escribir el signo (-) y finalmente podemos escribir a cuantos pasos podría avanzar el objeto, simplemente con cambiar el 10 por un número de nuestra elección.

“girar 15 → grados”: Este bloque hace que nuestro objeto gira hacia la derecha, y ese valor de giro está dado en grados.

“girar 15 ← grados”: Este bloque funciona exactamente igual que el anterior, su diferencia es que gira en lado contrario a las manecillas del reloj.

“apuntar en dirección 90”: Este bloque hace que nuestros objetos “observen” en un ángulo de nuestra elección. Cuando se hace clic en “90”, saldrá una ventana emergente que nos brinda 4 opciones: (90) hacia la derecha, (-90) hacia la izquierda, (0) hacia arriba, (180) hacia abajo.

“apuntar hacia”: Este bloque nos da la oportunidad de que nuestro objeto gira en la posición donde se encuentre el cursor del ratón.

“ir a X:269 Y:1”: Este bloque tiene la opción de escribirle coordenadas en el eje X y en el eje Y, para luego posicionar el objeto en esa posición.

“ir a”: Si queremos que el objeto siga la posición del cursor, solo basta con usar este bloque.

“deslizar en 1 seg a X:269 Y:1”: Este objeto permite que nuestro objeto se desplace exponencialmente en segundos a alguna coordenada de nuestra elección. El tiempo de desplazamiento puede ser escrito, cambiando el valor 1 por defecto por el que gustemos.

“cambiar X por 10”: Este bloque permite que nuestro objeto se desplace en el eje X. Si queremos movernos hacia la derecha, el valor debe estar positivo, si queremos movernos hacia la izquierda, un valor negativo.

“fijar X a 0 ”: Este bloque fija una coordenada para el eje X, teniendo en cuenta que si queremos una posición positiva, el valor debe estar positivo, una posición negativa, un valor negativo.

“cambiar Y por 10”: Este bloque permite que nuestro objeto se desplace en el eje Y. Si queremos movernos hacia arriba, el valor debe estar positivo, si queremos movernos hacia abajo , un valor negativo.

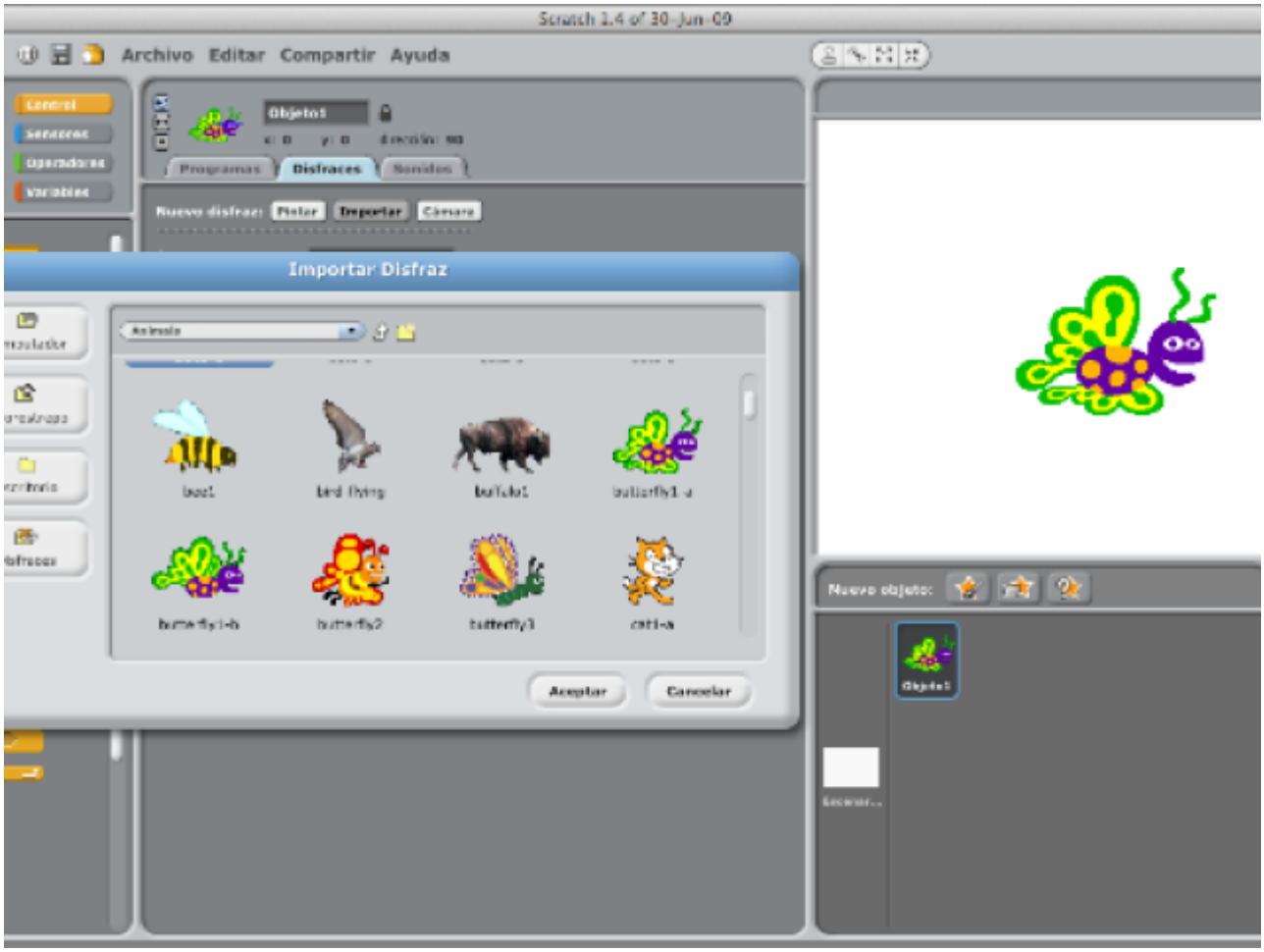
“fijar Y a 0”: Este bloque fija una coordenada para el eje Y, teniendo en cuenta que si queremos una posición positiva, el valor debe estar positivo, una posición negativa, un valor negativo.

“rebotar si está tocando un borde”: Este bloque debe ser usado junto con un bloque de control. Cuando el objeto toque algún borde del escenario, el tomara rebotara hacia la posición contraria.

Los bloques “posición en x, posición en y, y dirección” : nos sirven para comparar la posición de un objeto junto con una variable. Lo veremos más adelante.

Ejercicio usando los bloques vistos.

Crearemos un escenario que sea parecido a un jardín, puedes importarlo u puedes dibujarlo en la interfaz de Scratch.

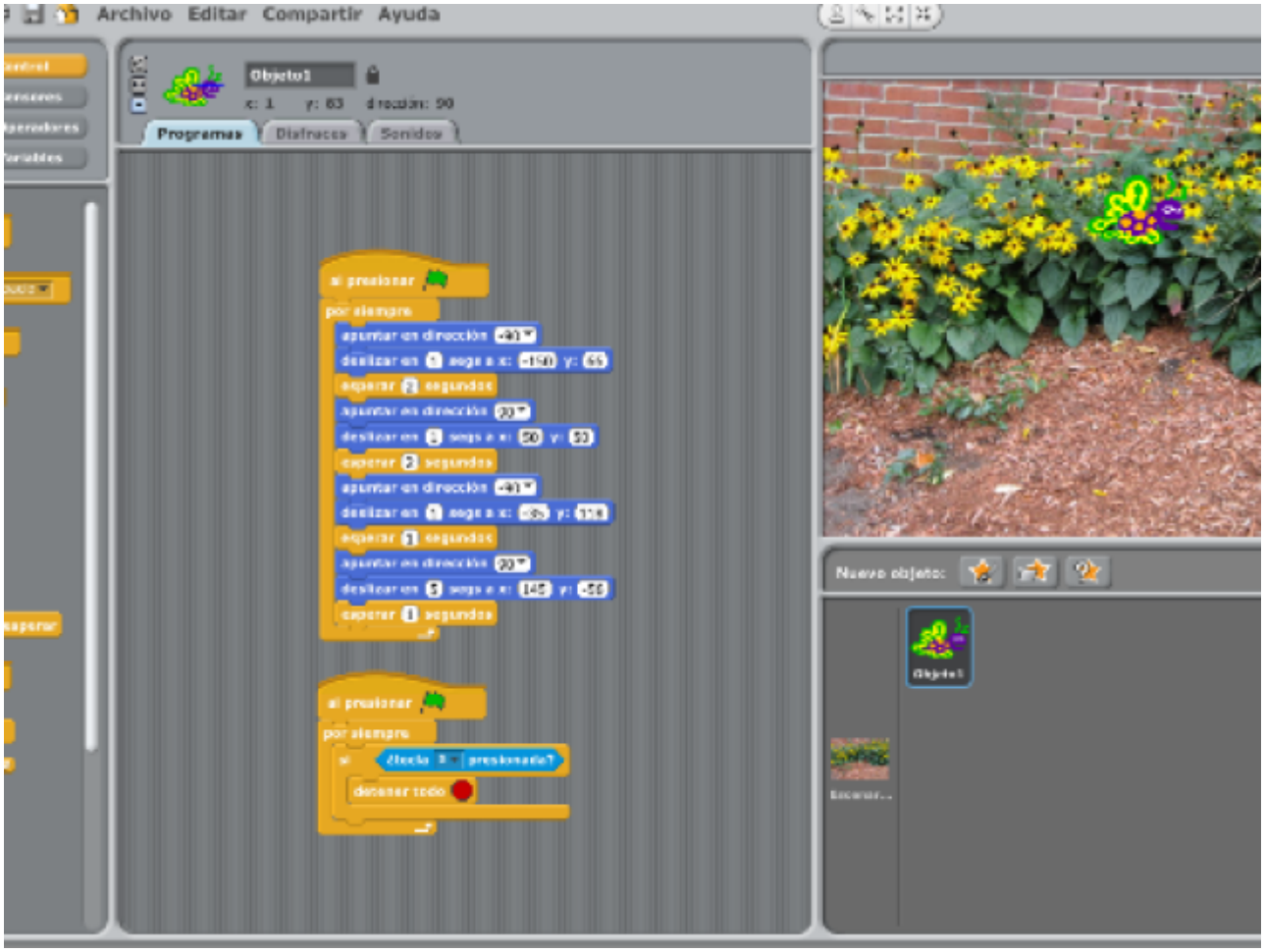


Luego nuestro personaje será una mariposa que deambula por el jardín, ella estará en una flor y se quedará unos segundos allí, luego cambiara de posición y así se la pasara. El programa debe detenerse cuando presionemos alguna tecla del computador.

Nuestro script debe verse parecido a esta imagen.

El fondo, las flores, es una imagen que elegimos en la biblioteca de scratch.

Luego elegimos un personaje (una abeja) y a esta imagen, realizando el script de la siguiente imagen a nuestro personaje.

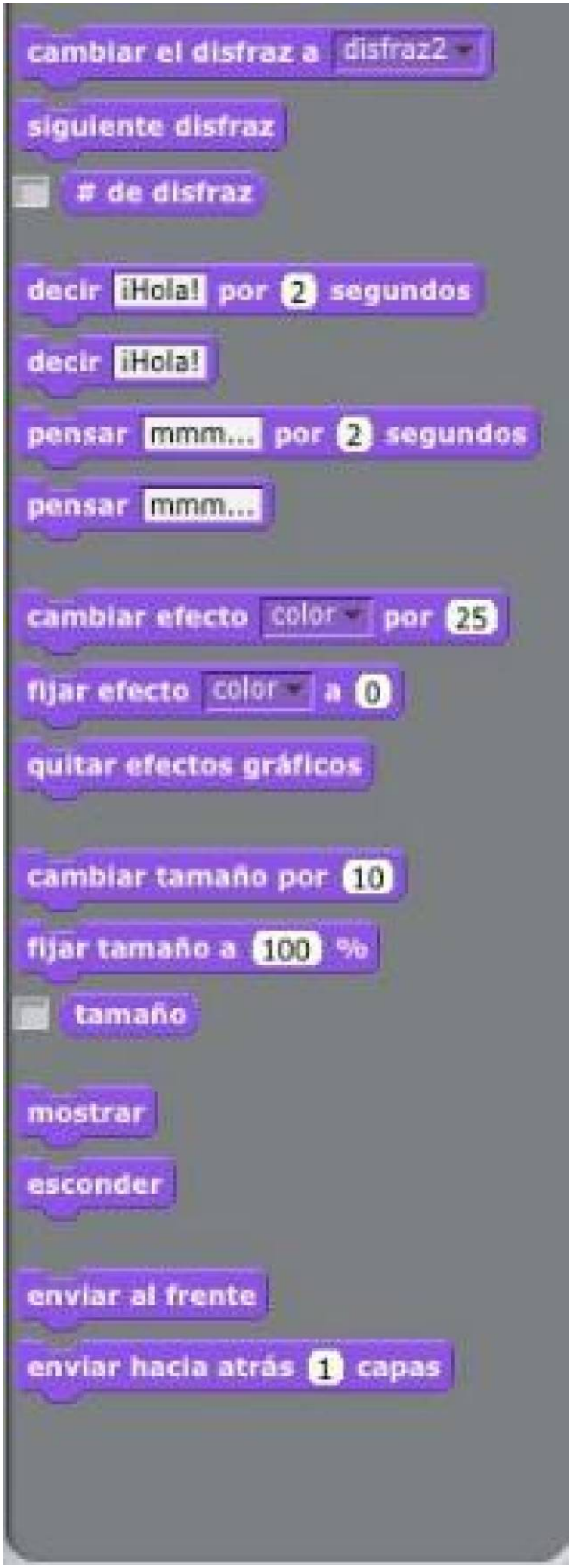


El siguiente programa se encuentra disponible para su descarga en: Enlace al archivo

Recapitulando: Explicamos para que servían los bloques contenidos en las secciones de **Control y Movimiento;** recordemos que la sección de control contiene bloques que os permite hacer que nuestro programas tomen decisiones “lógicas” en el programa y la sección de Movimiento nos permite desplazar objetos en el escenario.

A continuación veremos las secciones de apariencia, sensores (muy importantes) y sonido.

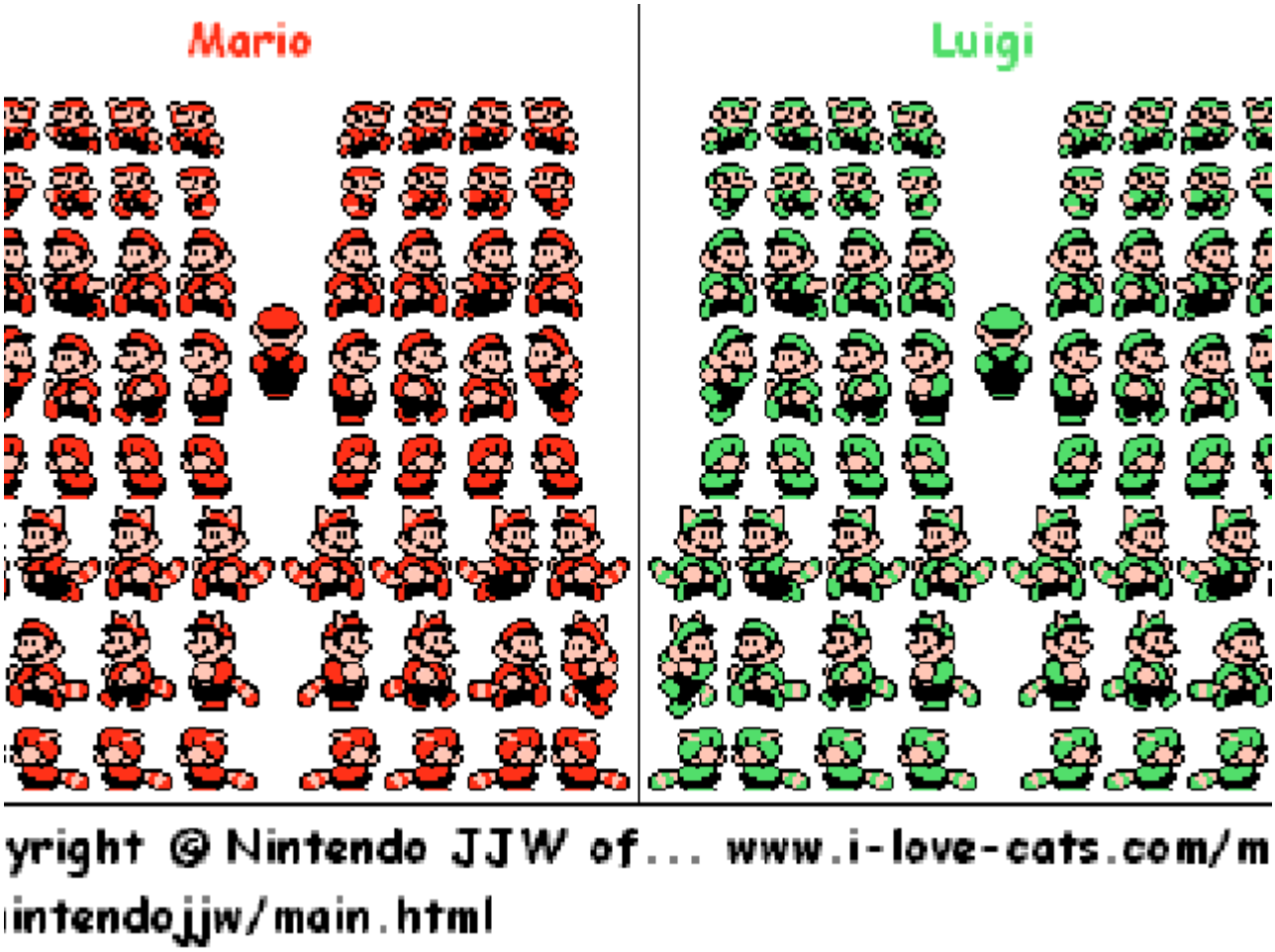
Sección de apariencia.


cambiar el disfraz a disfraz2
siguiente disfraz
de disfraz
decir ¡Hola! por 2 segundos
decir ¡Hola!
pensar mmm... por 2 segundos
pensar mmm...
cambiar efecto color por 25
fijar efecto color a 0
quitar efectos gráficos
cambiar tamaño por 10
fijar tamaño a 100 %
tamaño
mostrar
esconder
enviar al frente
enviar hacia atrás 1 capas

A grandes rasgos, en la sección de apariencia tenemos la posibilidad de escribirle a nuestras imágenes mensajes tipo historietas, cambiarles el color, agregar efectos y finalmente cambiar entre Sprites.

Un Sprite son generalmente imágenes secuenciales de una acción generalmente de algún personaje, que se usan para realizar animaciones por computador.

En la siguiente imagen podemos observar una serie de Sprites usados y diseñados en la saga de **Super Mario Bros.** Generalmente los Sprites son usados para crear movimiento en una imagen.



¿ Que hace cada bloque de la sección de apariencia ?

Cambia la apariencia del Sprite a diferentes disfraces segun esten en la pestaña **disfraces**

Hace un recorrido de la lista de Sprites contenidos en un objeto. Cuando llega al final, vuelve a comenzar, mira el ejemplo en esta imagen.

Indica el número del Sprite actual del objeto.

Muestra un mensaje dentro de una burbuja en el Sprite por cierta cantidad de tiempo.

Muestra un mensaje dentro de una burbuja en el Sprite. (se puede eliminar la burbuja del mensaje mediante la ejecución de este bloque sin texto.)

Muestra un mensaje en forma de pensamiento por cierta cantidad de tiempo.

Muestra un mensaje en forma de pensamiento en el Sprite.

Aplica a un Sprite un efecto visual contenido en este modulo y lo multiplica por una cantidad. (Las opciones se obtienen al desplegar la página.)

Parecido al módulo anterior, este módulo establece un efecto a un Sprite. La cantidad de efecto será elegida por un número.

Remueve todos los efectos gráficos contenidos en un Sprite.

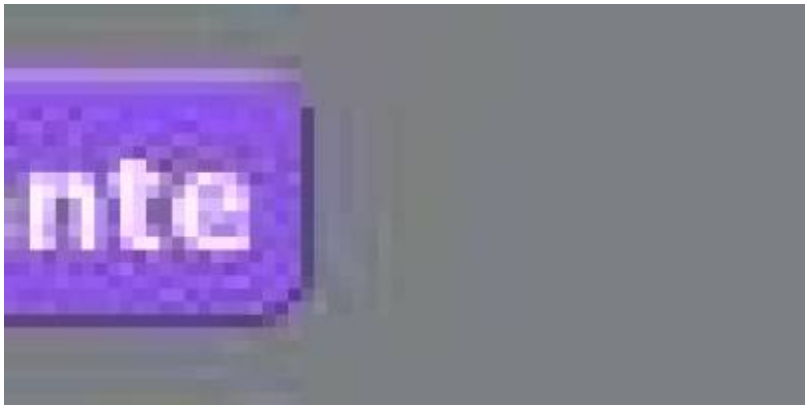
Cambia el tamaño del Sprite especificado por un número.

Cambia el tamaño original del Sprite a un % especificado.

Informa del tamaño del Sprite, como un % del original.

Muestra los Sprites en el escenario.

Esconde los Sprites del escenario.

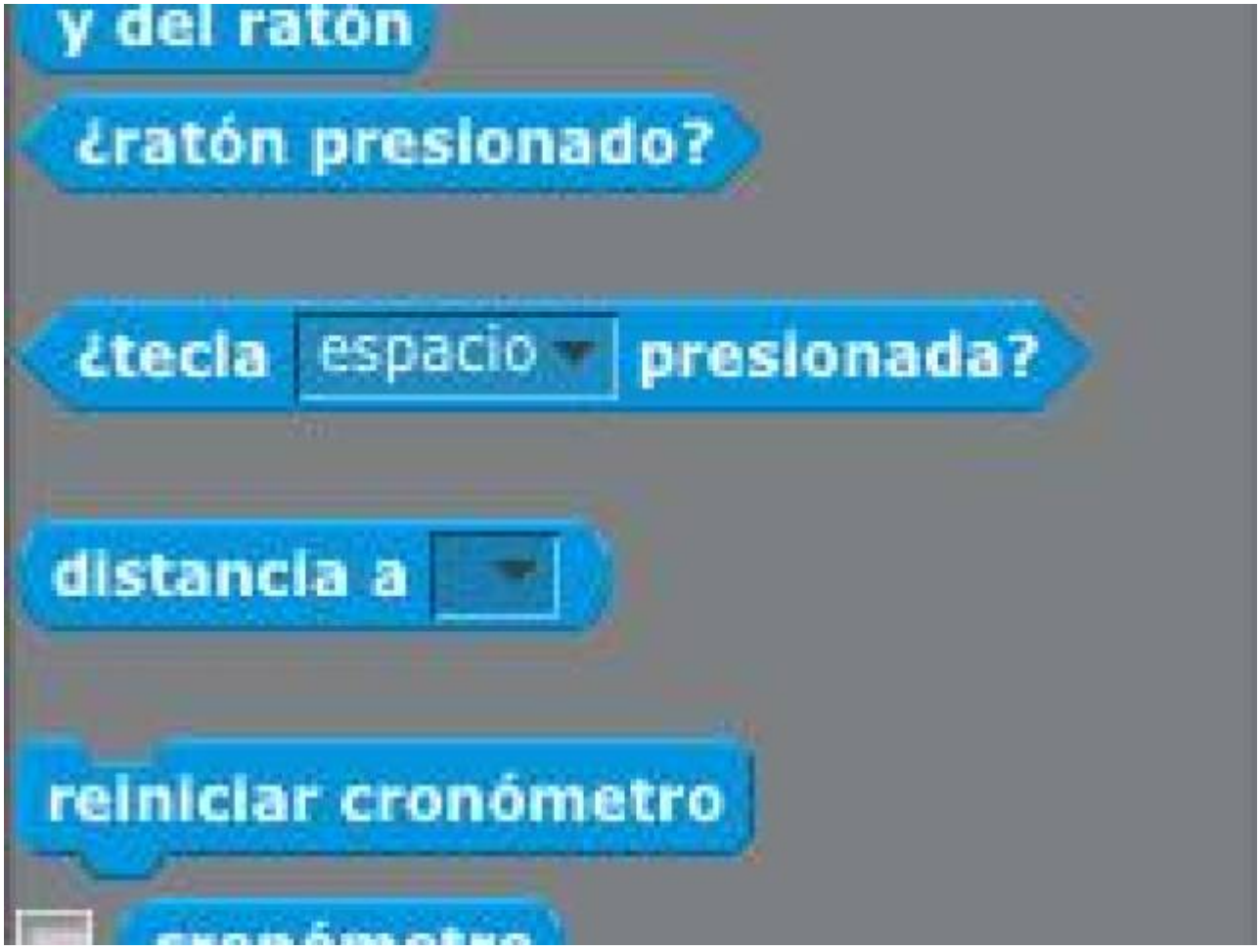


El objeto con este bloque es enviado hacia el frente de todos los Sprites.

Mueve el Sprite a un número especificado de capas, de modo que pueda estar oculto detrás de otros Sprites.

Sección de sensores.

En esta sección encontramos bloques que complementan la funcionalidad de la sección de los bloques de control. Están diseñados para encajar en la entrada de algunos bloques y los reconoceremos porque sus extremidades forman un triángulo.

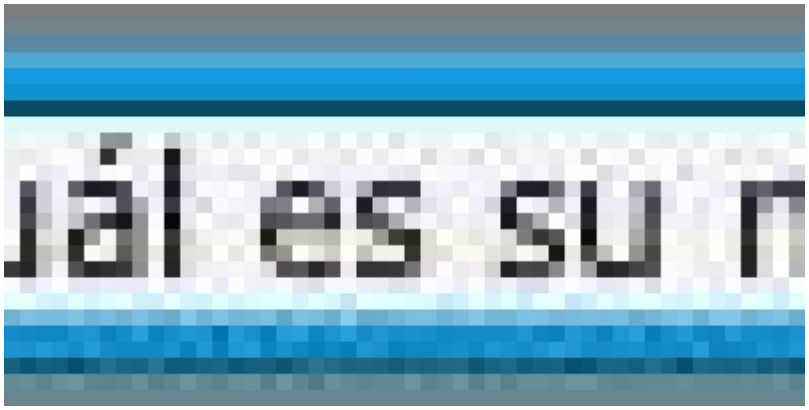


¿ Que hace cada bloque de la sección de sensores ?

Este bloque arroja verdadero si el Sprite hace contacto con un borde del escenario o con el puntero del ratón. Las opciones se pueden desplegar haciendo clic en el icono de la flecha hacia abajo.

Este bloque arroja verdadero si el Sprite está tocando un color específico. Para seleccionar la muestra de color, clic en la muestra de color, a continuación, utilizar cuentagotas para seleccionar el color.

Este bloque arroja verdadero si el primero color (Dentro del Sprite) está tocando algún otro color de otro Sprite o del escenario. Para seleccionar la muestra de color, clic en la muestra de color, a continuación, utilizar cuentagotas para seleccionar el color.



Hace una pregunta en pantalla, la cual puede ser escrita en el espacio. Luego el teclado la almacena y luego hace que el programa espere hasta que se presione la tecla Enter.

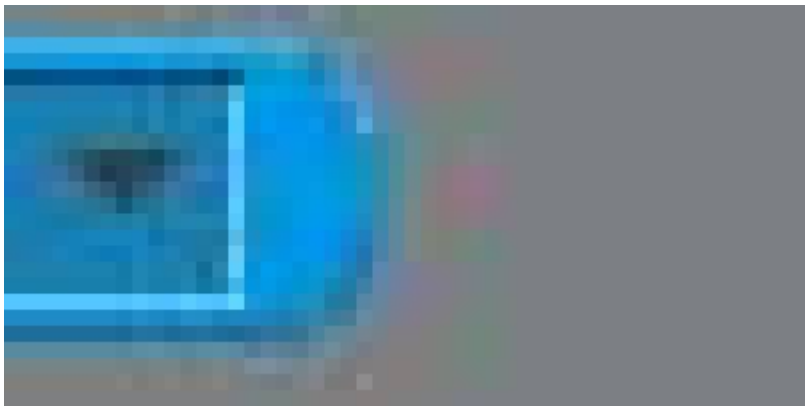
Informa sobre la última escritura en el teclado usando. **Y esta información es usada por todos los Sprites.**

Informa la posición en x del puntero de ratón.

Informa la posición en y del puntero de ratón.

Reporta verdadero si el ratón es presionado.

Reporta verdadero si una tecla de nuestro computador es presionada.



Informa la distancia entre un Sprite especificado y el puntero del ratón.

Establece el contador a cero. Más adelante explicaremos con un ejemplo.

Muestra el valor del temporizador en segundos.

El temporizador siempre está corriendo.

Informa una propiedad o variable de otro elemento Sprite.

Informa el volumen (de 1 a 100) de los sonidos detectados por el micrófono de la computadora.

Informa verdadero si el micrófono de la computadora detecta un volumen de sonido más fuerte de 30 (en una escala de 1 a 100).

Reporta el valor del sensor especificado. Para utilizar este bloque, se necesita un sensor conectado a su ordenador. usted puede usar esto con una placa de sensores de scratch mas info o con LEGO ® WeDoTM mas info

Informa verdadero si el sensor se especifica que se presiona. Para utilizar este bloque, es necesario tener una tarjeta de sensor arañazos conectado a su ordenador. mas info

Sección de sonido

En la sección de sonido podemos encontrar bloques que nos permitirán añadir sonidos a nuestros scripts. Ya sea por medio de notas musicales, sonidos que tengamos almacenados en el computador o finalmente archivos que grabemos en el momento.

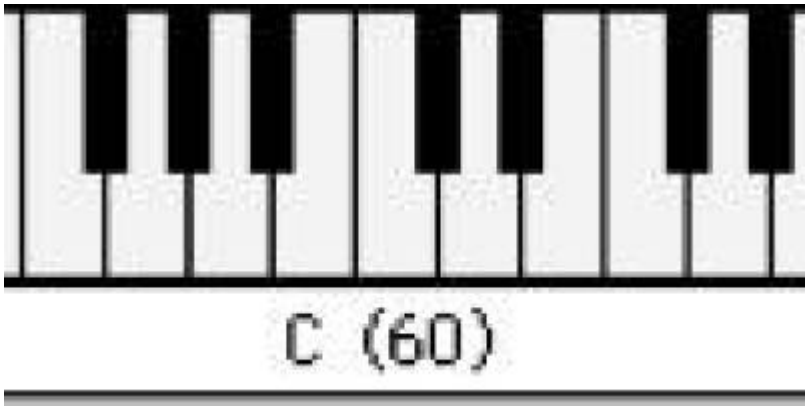
¿ Que hace cada bloque de ****la sección de sonido ?

Inicia la reproducción de un sonido, seleccionado en el menú desplegable, e inmediatamente pasa al siguiente bloque incluso cuando el sonido se está reproduciendo. Por lo general, siempre está el sonido miau del gato de scratch.

Reproduce un sonido y espera hasta que el sonido termine de sonar, para continuar con el siguiente bloque.

Detiene todos los sonidos.

Reproduce un sonido de batería, seleccionado en el menú desplegable, por un determinado tiempo.



Reproduce una nota musical para una nota especificada por un tiempo determinado.

Establece un instrumento musical o sonido para cada Sprite.

Cambia el volumen de un sonido a una cantidad específica.

Establece el volumen de un sonido de un Sprite específico.

Muestra la cantidad de volumen de un sonido.

En música el tiempo de una canción es regido por el tempo. Este bloque permite acelerar o ralentizar nuestro tempo. Trabajaremos en este bloque más adelante.

Este bloque establece un tempo para nuestra secuencia musical.

Informa en tempo en BPM (beats por minuto).

Empezamos ya con la última parte de los bloques, que nos permiten escribir programas en Scratch. Veremos dos muy importantes, que serán los operadores que complementan muchos los bloques de control junto con los de movimiento y finalmente una parte muy especial llamado Variables, el cual nos permitirá crear nuestros propios valores cambiables , esto por ejemplo sirve para almacenar las vidas de un jugador, indicar las vidas del enemigo y para mil cosas más.

En el apartado lápiz podemos encontrar bloques que permitirán dibujar a los Sprites en el programa, son muy útiles sobre todo para dibujar nuestras propias formas en el lienzo del programa. Hablaremos de ella en breve:

Sección de Operadores.

En la sección de operadores podemos encontrar bloques que nos permiten comparar información con otra información. Estas opciones pueden estar en Mayor que (>), Menor que (<) o igual que (=); podemos sumar, restar, multiplicar o dividir números. Podemos encontrar muchos bloques pero los descritos anteriormente son, algunos de ellos. Veamos su descripción más detalladamente.

¿ Que hace cada bloque de la sección de operadores ?

Permite sumar dos números, los números son escritos en los espacios en blanco.

Permite restar dos números, , los números son escritos en los espacios en blanco.

Permite multiplicar dos números, , los números son escritos en los espacios en blanco.

Permite dividir dos números, , los números son escritos en los espacios en blanco.

Toma un número entero y genera números aleatorios dentro de una rango especificado.

Informa verdadero si el primer valor es menor que el segundo.

Informa verdadero si los dos valores son iguales.

Informa verdadero si el primer valor es mayor que el segundo.

Informa verdadero si ambas condiciones son verdaderas.

Informa verdadero si una u otra condición es verdadera.

Reporta verdadero si la condición es falsa, informa falso si la condición es verdadera.

Combina dos palabras.

Informa la posición de la letra de una palabra especificada.

Informa el número de letras en una cadena.

Informa el residuo de la división del primer número con el segundo número.

acerca un número decimal a un número entero; por ejemplo, 5.8 redondeado 6.

Informes el resultados de la función seleccionada (abs, sqrt, sin, cos, tan, asin,

acos, atan, ln, log, e ^, 10 ^) aplicada al número especificado.

Sección de Lápiz.

A rasgos generales, los bloques contenidos en la sección de lápiz sirven para hacer dibujos en nuestros escenarios.

¿ Que hace cada bloque de la sección de lápiz ?

Borra todas las marcas de lápiz y sellos del escenario.

Coloca abajo el lápiz del Sprite , por lo que se dibuja a medida que avanza. Este bloque generalmente se usa para que nuestras animaciones usando un Sprite dibujen en el escenario.

Saca la pluma del Sprite, por lo que no dibujara a medida que avanza.

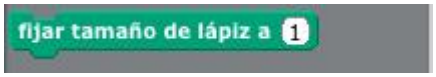
Fija un color para nuestro lápiz, seleccionandolo en la casilla, esto también determina la sombra de nuestro lápiz.

Cambia el color del lápiz por una cantidad especificada.

Define el color del lápiz al valor especificado.

Cambia la intensidad del lápiz por un valor específico, este valor normalmente suele llamarse la opacidad o transparencia del color.

Cambia el grosor del lápiz.



Establece el grosor del lápiz a un valor fijo.

sella el escenario con diversas imágenes.

Sección de variables.

Por lo general cuando cliqueamos en la sección de variables, no encontramos bloques para elegir. Solo dos opciones están dispuestas para empezar a crear nuestros bloques.

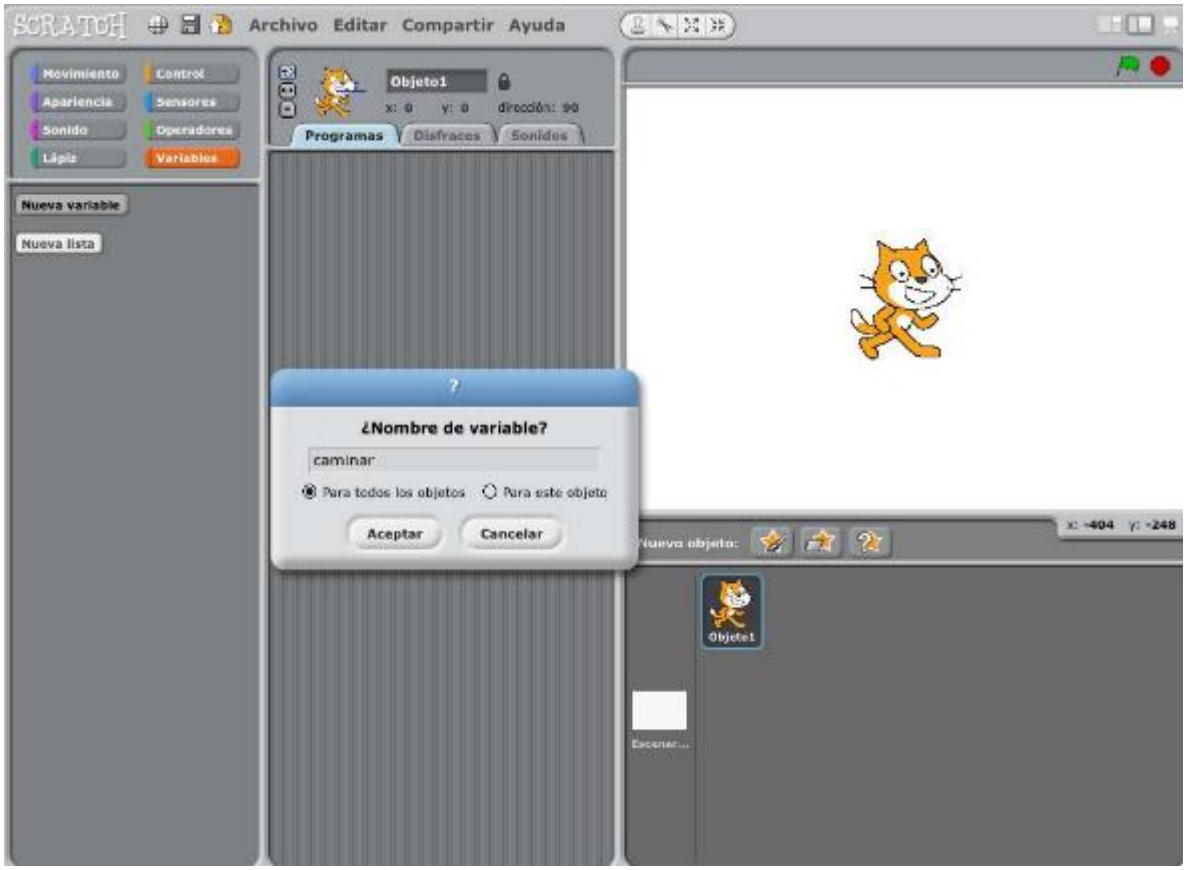
Con el primer botón “nueva variable” podemos crear bloques y funcionalidades que podrán cambiar con el paso del programa. A estos bloques los llamaremos “variables”.

En el segundo botón, “Nueva lista” podemos crear y nombrar una nueva lista. Al crear una lista, por primera vez, aparecerán los elementos de la lista. podemos elegir si la lista es para todos los Sprites (global) o simplemente para una Sprite (local).

Para crear nuevas variables o nuevas listas simplemente cliqueamos sobre una de ellas y elegimos el nombre que deseamos darle, además de si será local o global.

Cuando se dice que una variable es local, solo aplicara al objeto en cuestión, en este caso aplica solo al sprite. Pero si por el contrario la variable de denomina como global, esta puede ser aplicada y

usada en varios objetos a la vez.



Luego de este paso, aparecerán una serie de objetos que podremos utilizar en la elaboración del script.

¿ Que hace cada bloque de variables ?

Para nuestro ejemplo, la variable se llama caminar. Entonces explicaremos con este ejemplo el significado de cada bloque.

¿Que hace cada bloque después de creada la variable ?

Borra todos los bloques asociados con esta variable.

Muestra el valor de la variable.

Establece la variable “caminar” por un valor específico.

Cambia la variable “caminar “ por una cantidad especificada. Si tiene más de una variable, es conveniente utilizar el menú desplegable para seleccionar el nombre de la variable.

Muestra en la pantalla el estado de las variables.

Esconde de la pantalla el estado de las variables.

Sección de listas.

Las listas funcionan como las variables, las cuales pueden almacenar números, así como cadenas de letras y otros caracteres.

Para crear una lista se debe hacer clic en el nombre de la lista y luego nombrarla.

Al crear una lista por primera vez, aparecerán los elementos de la lista; se puede elegir si la lista es para todos los Sprites (global) o simplemente para una Sprite (local).

¿ Que hace cada bloque de las listas ?

Para nuestro ejemplo, la variable se llama caminar. Entonces explicaremos con este ejemplo el significado de cada bloque.

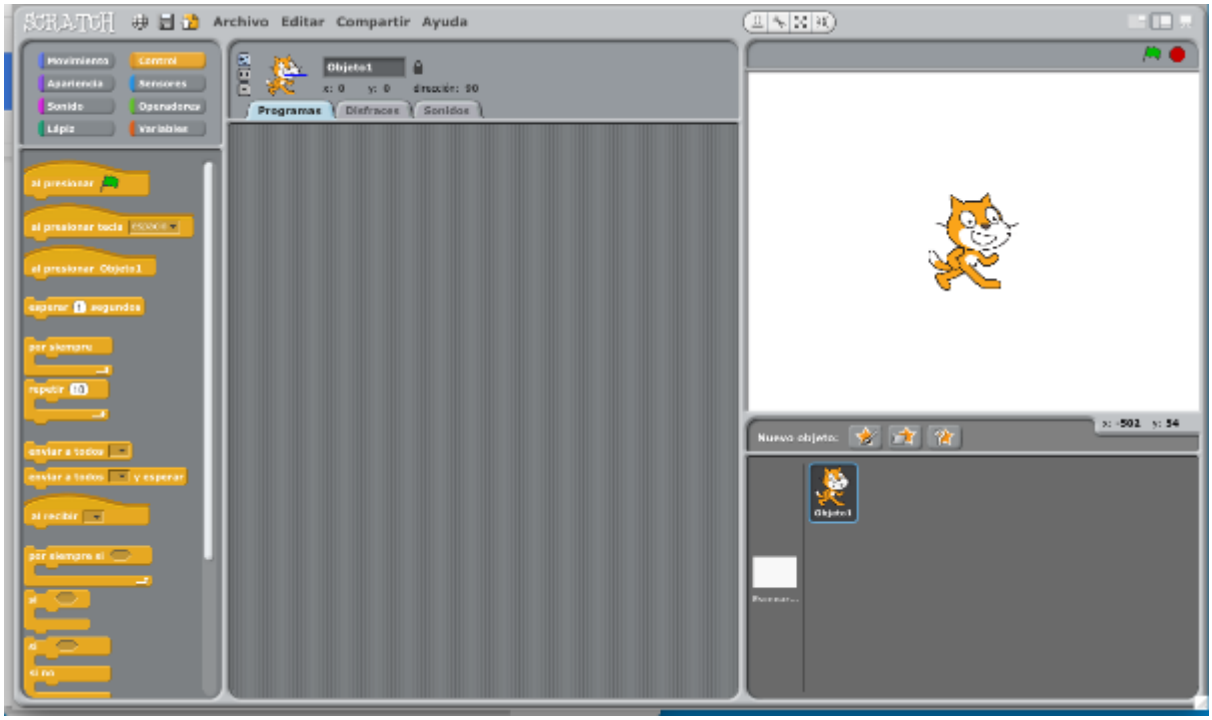
botón para crear una nueva lista.

Sección dos: Ejercicios prácticos

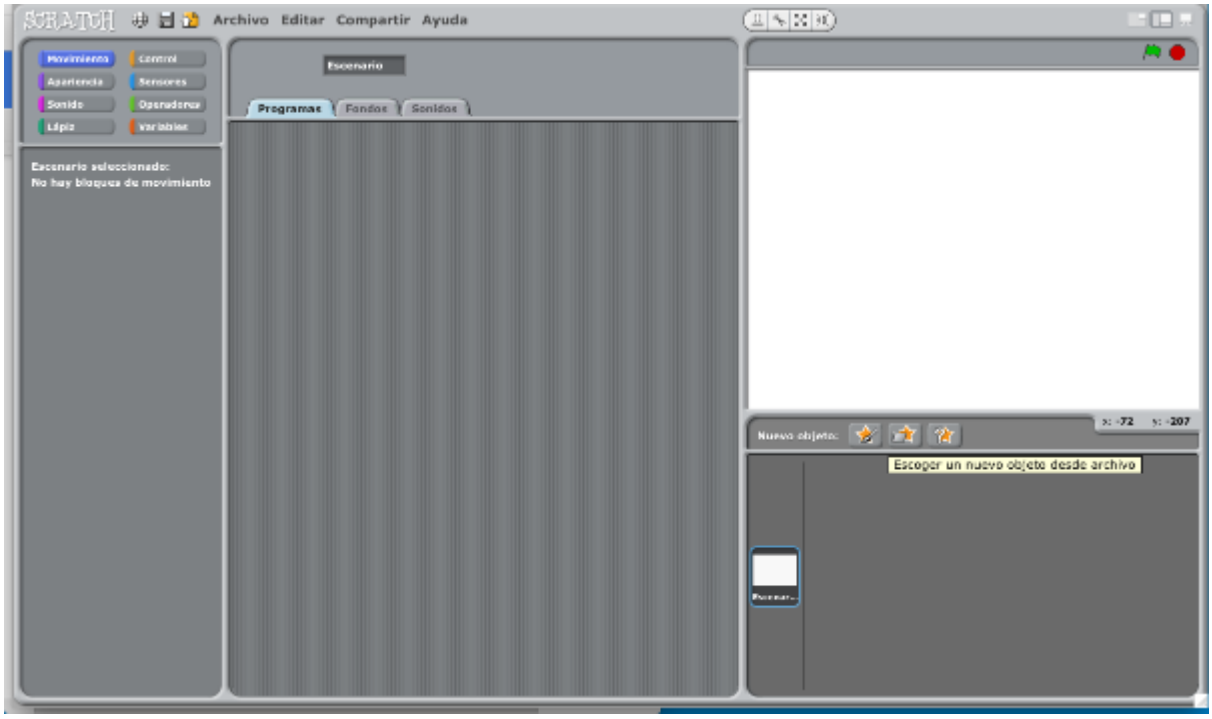
Animación simple a través de las teclas del teclado

Objetivo: la idea principal de este ejercicio inicial será conocer a través de un ejercicio simple como funciona la programación en scratch.

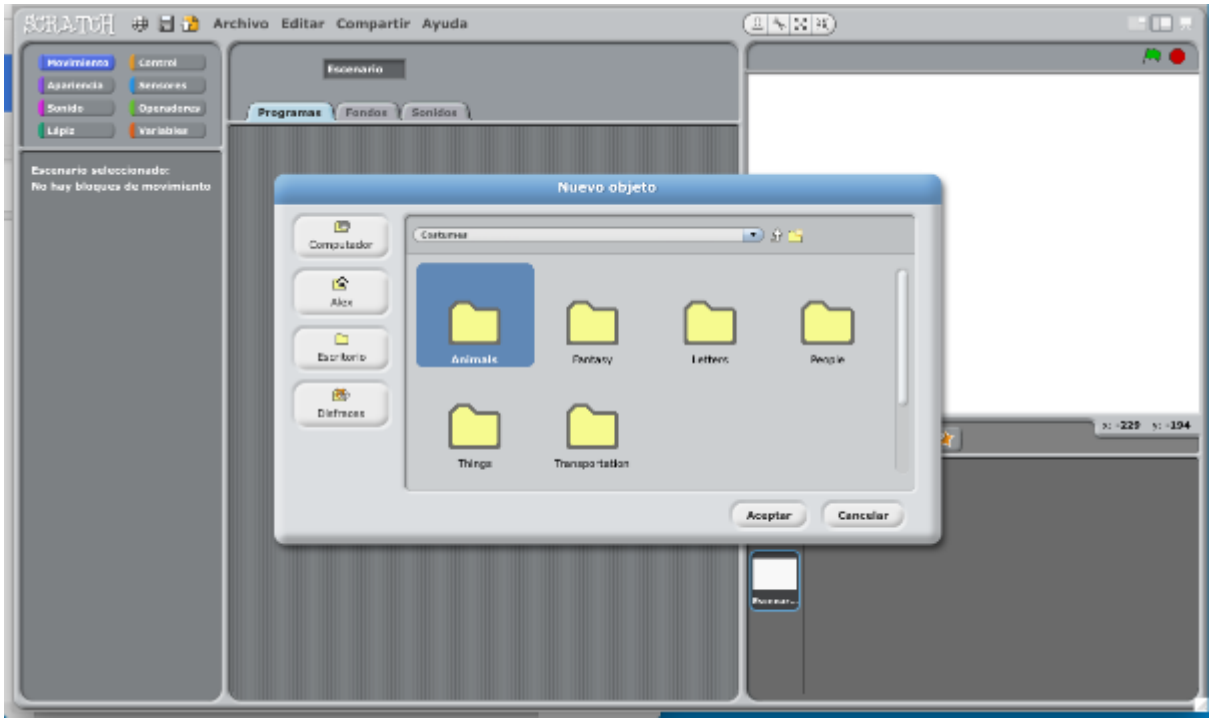
Iniciamos entonces con nuestra interfaz por defecto con el gato de scratch.



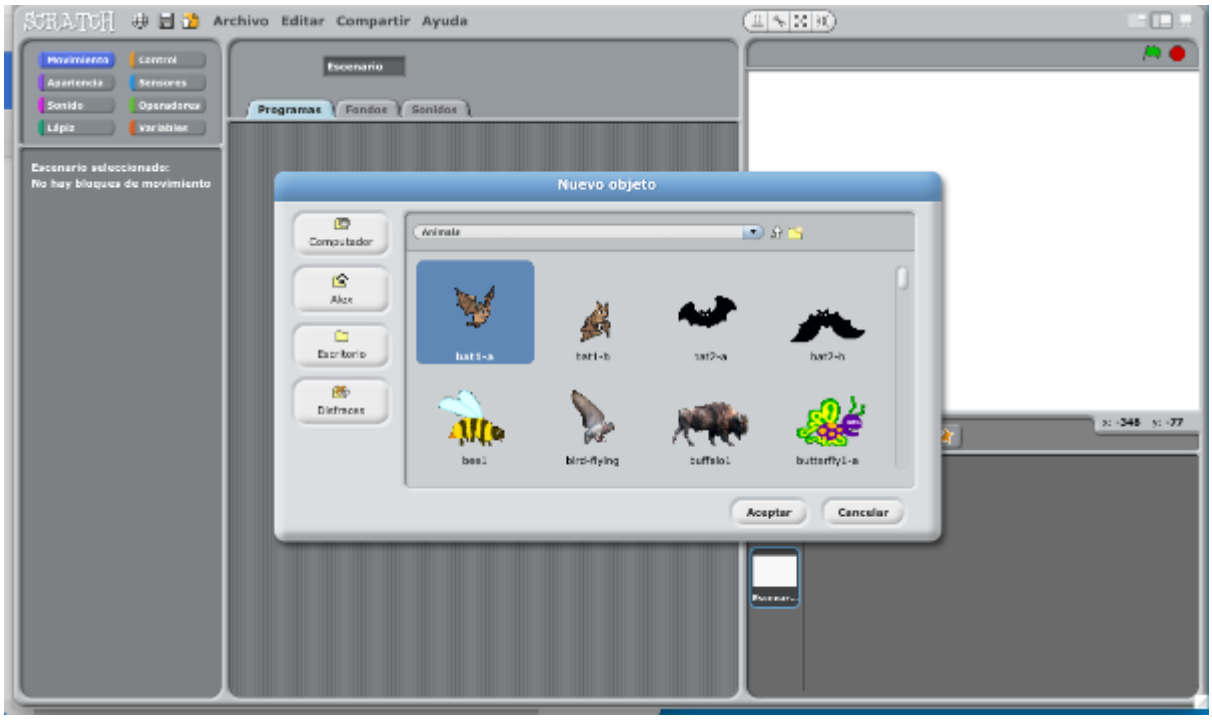
hacemos click derecho sobre el gato en la sección de objetos para borrarlo. Después hacemos click sobre la imagen de la estrella para importar un objeto nuevo.



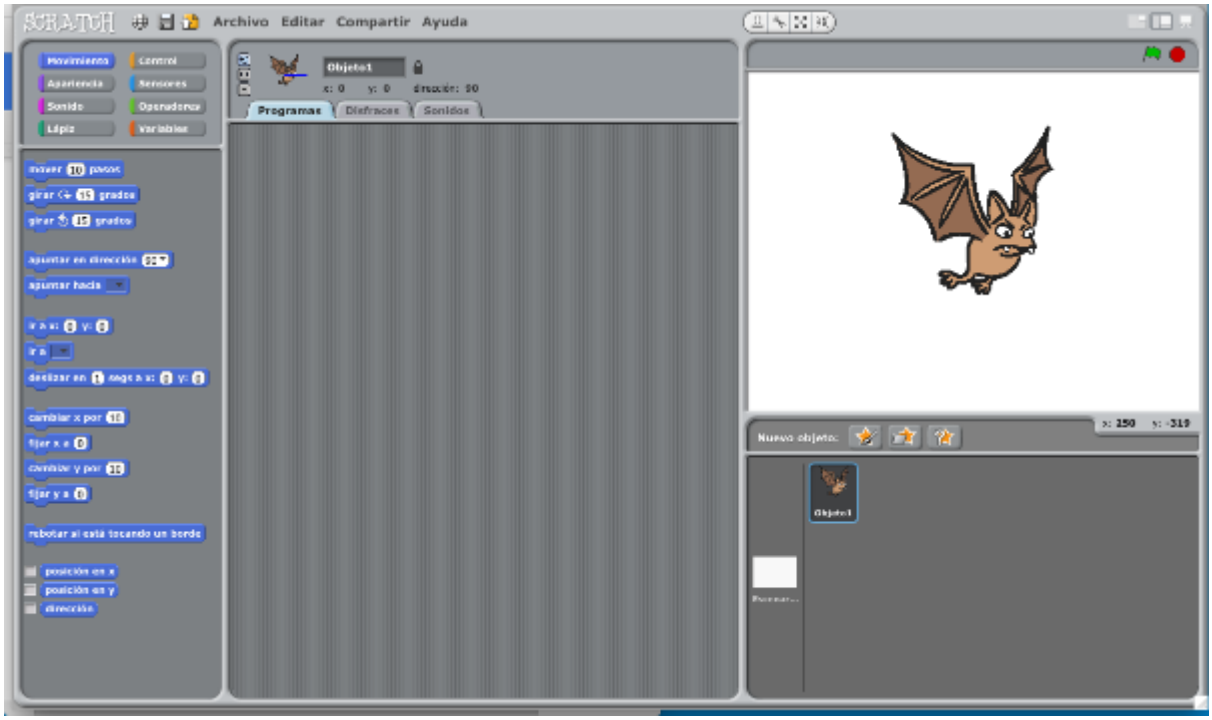
Cuando hagamos esto, el programa desplegará una nueva ventana con algunas carpetas. En las carpetas de la izquierda hacemos click donde dice disfraces y luego en la carpeta animales.



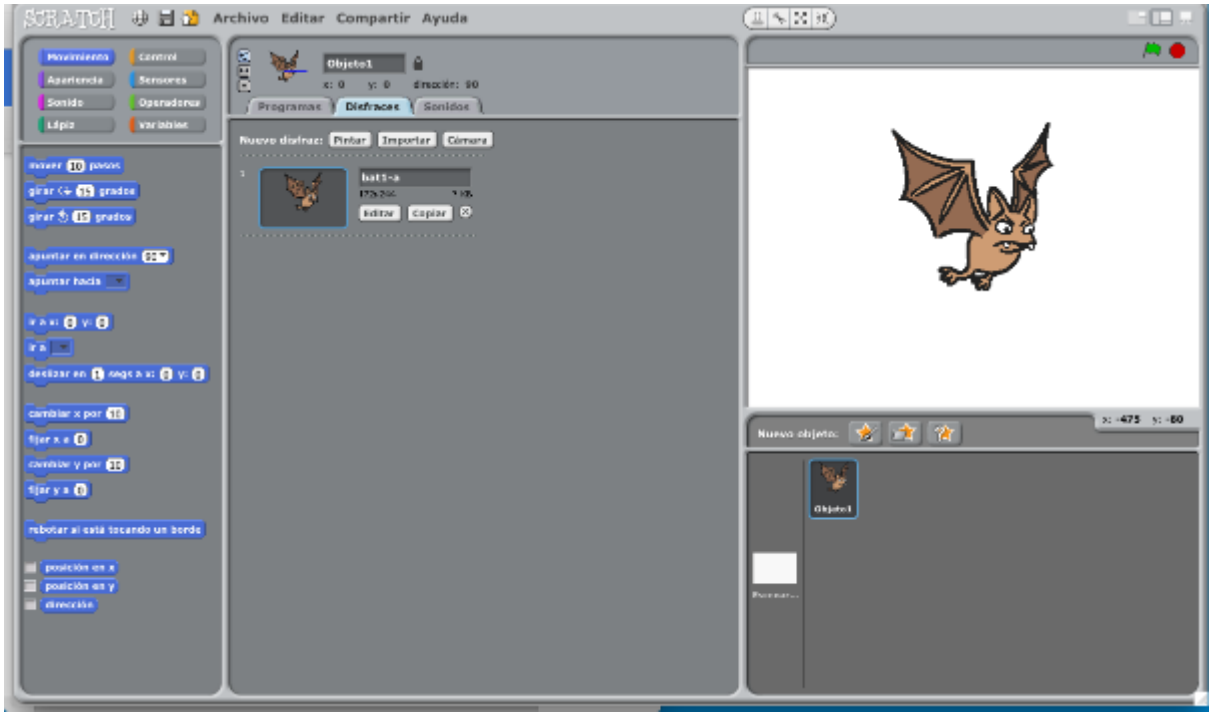
Dentro de ella podemos encontrar una infinidad de dibujos ya listos para trabajar. para este ejercicio en específico, trabajaremos con las dos sprites del murciélago. Donde una imagen este tiene las alas hacia arriba y en una imagen las alas hacia abajo.



Tras importar el primer murciélago, la ventana desaparecerá y la interfaz volverá a su estado normal mostrándonos la imagen seleccionada.

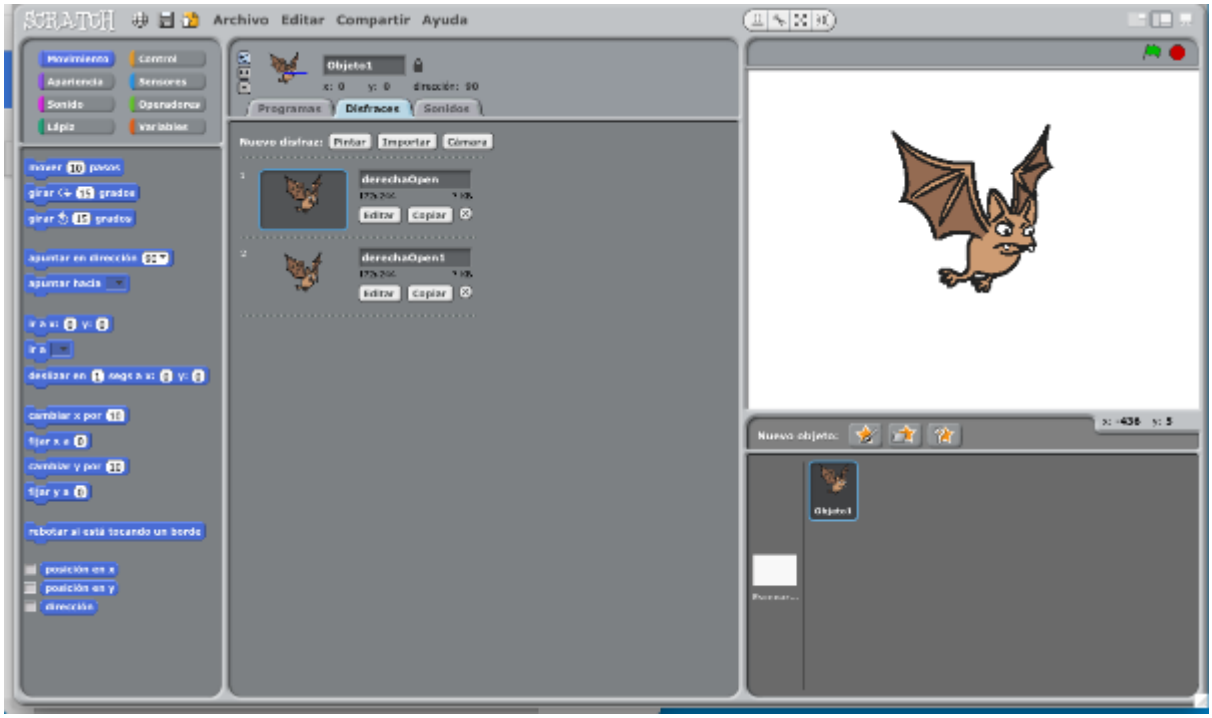


Tras este paso, en las pestañas de la sección de la mitad, damos click en disfraces. Donde la sección cambiará y se nos mostrará esto en pantalla.



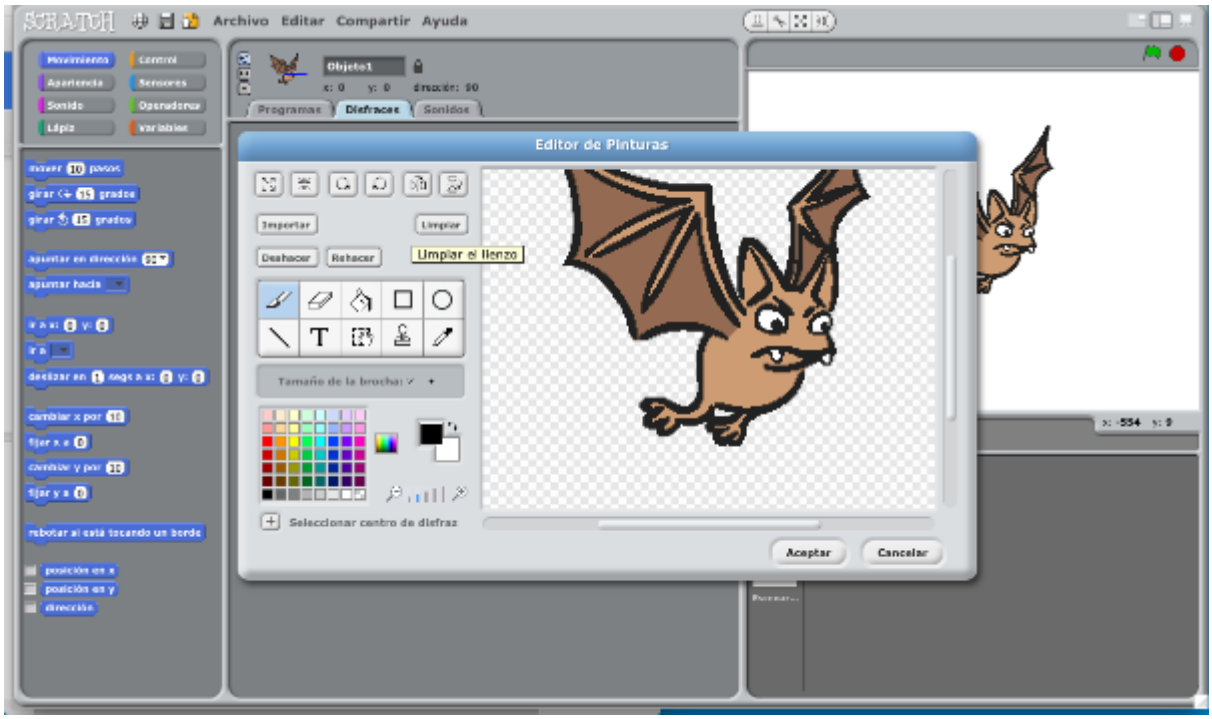
Aqui podemos darle un nombre al número del sprite de nuestra imagen. Lo cambiaremos a *derechaOpen* para saber que es la imagen del murciélago mirando hacia la derecha y con las alas abiertas.

Tras este paso, en la imagen hay dos botones inferiores que dicen editar y copiar. En el primero podemos editar la imagen para cambiarle el color o hacer modificaciones visuales. El otro que dice copiar hace una copia exacta del sprite original. Tras hacer click sobre copiar. La pantalla quedará así.

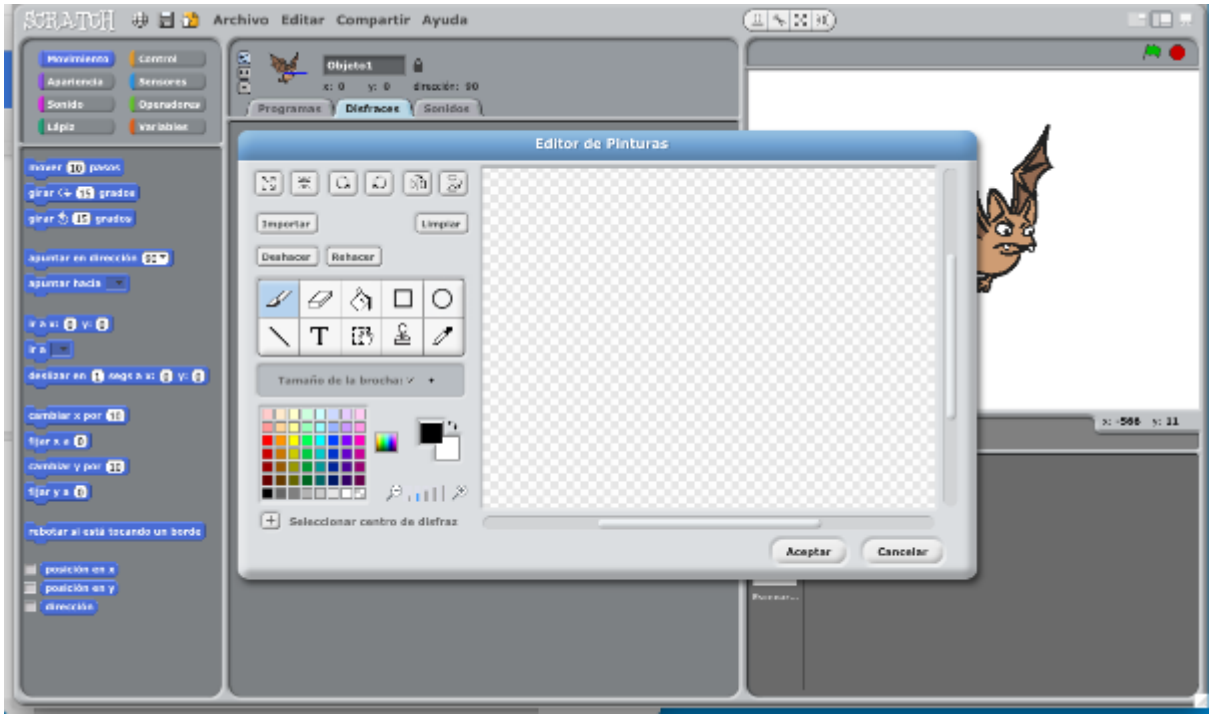


haremos click sobre la segunda imagen segunda imagen y haremos click en editar.

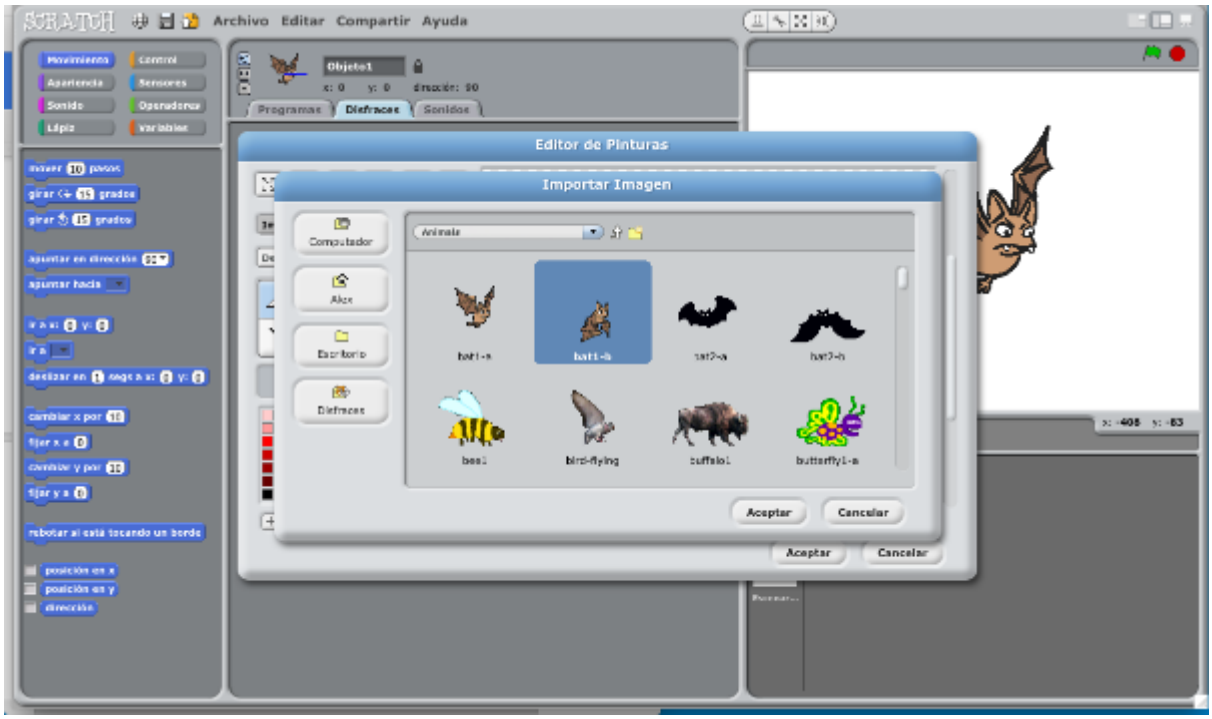
Tras hacer click en editar se nos abrirá una pantalla para dibujar imagenes con algunos botones y otras opciones.



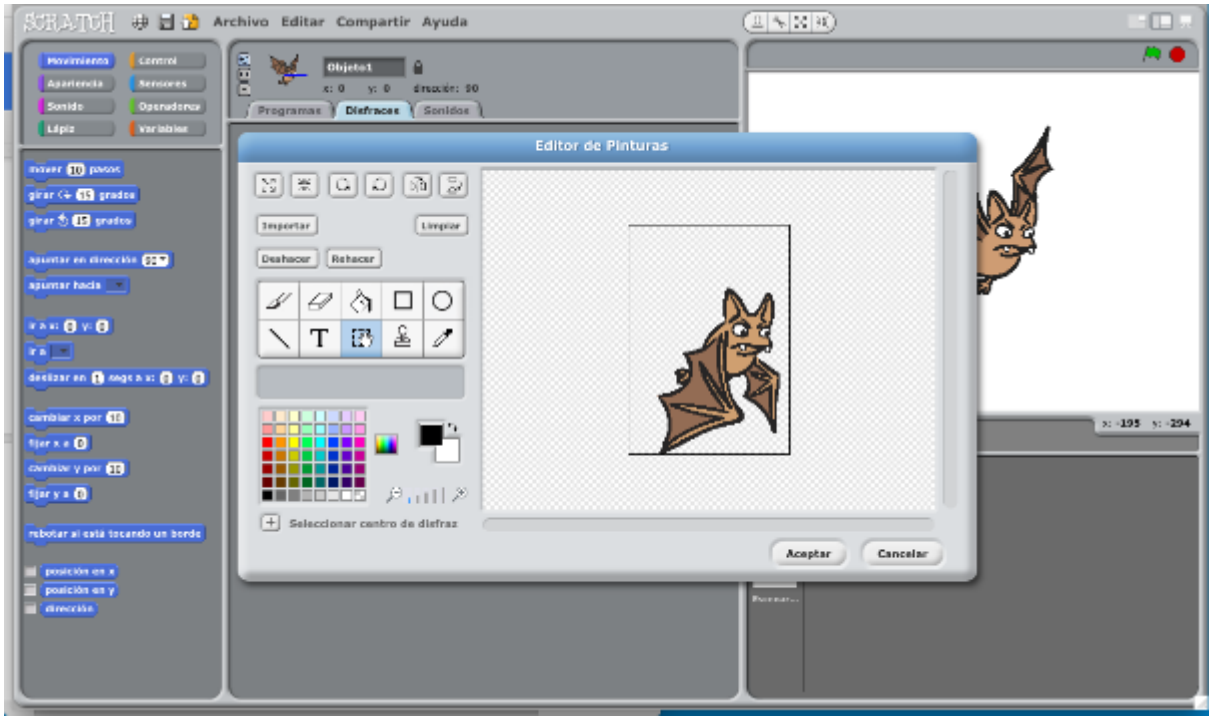
En la sección de herramientas de esta pantalla, hay un botón que dice limpiar. Tras hacer click sobre este botón; la pantalla de dibujo quedará limpia.



Haremos click sobre el botón importar y por defecto el programa abrirá la última carpeta que abrimos para importar una imagen al programa. En efecto, abrirá la carpeta de imágenes de Scratch y seleccionaremos la imagen del murciélago con las alas abajo.

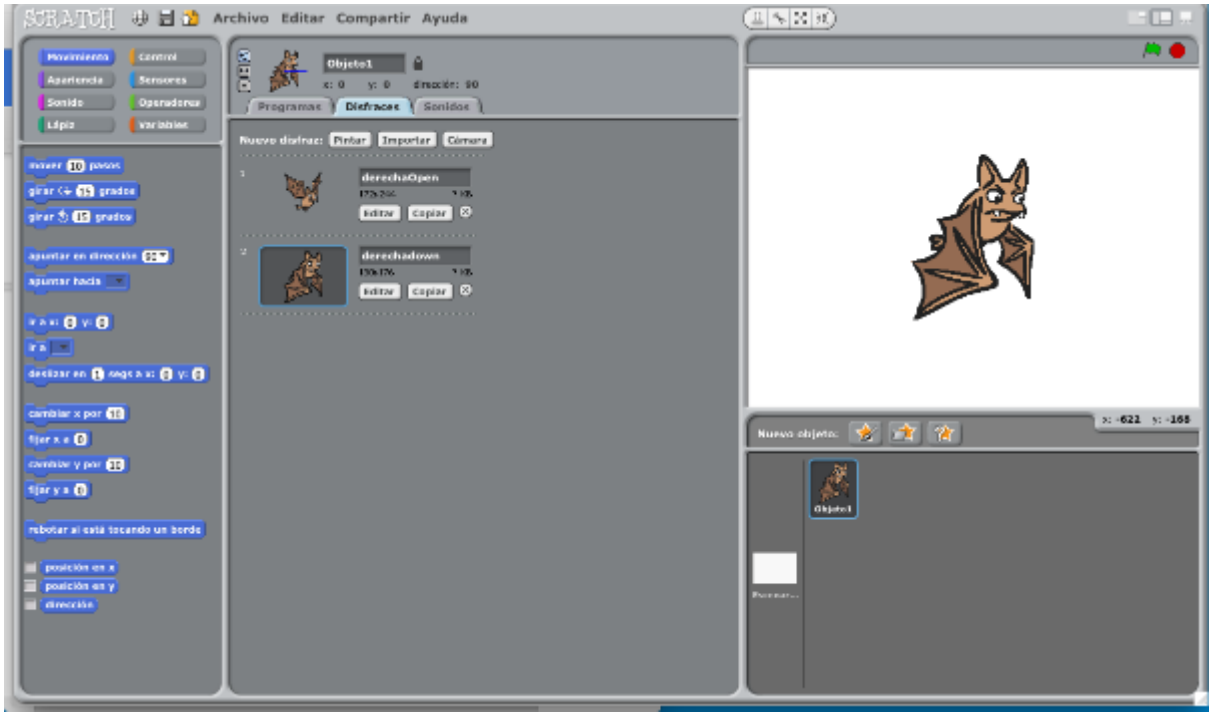


Tras importar la imagen

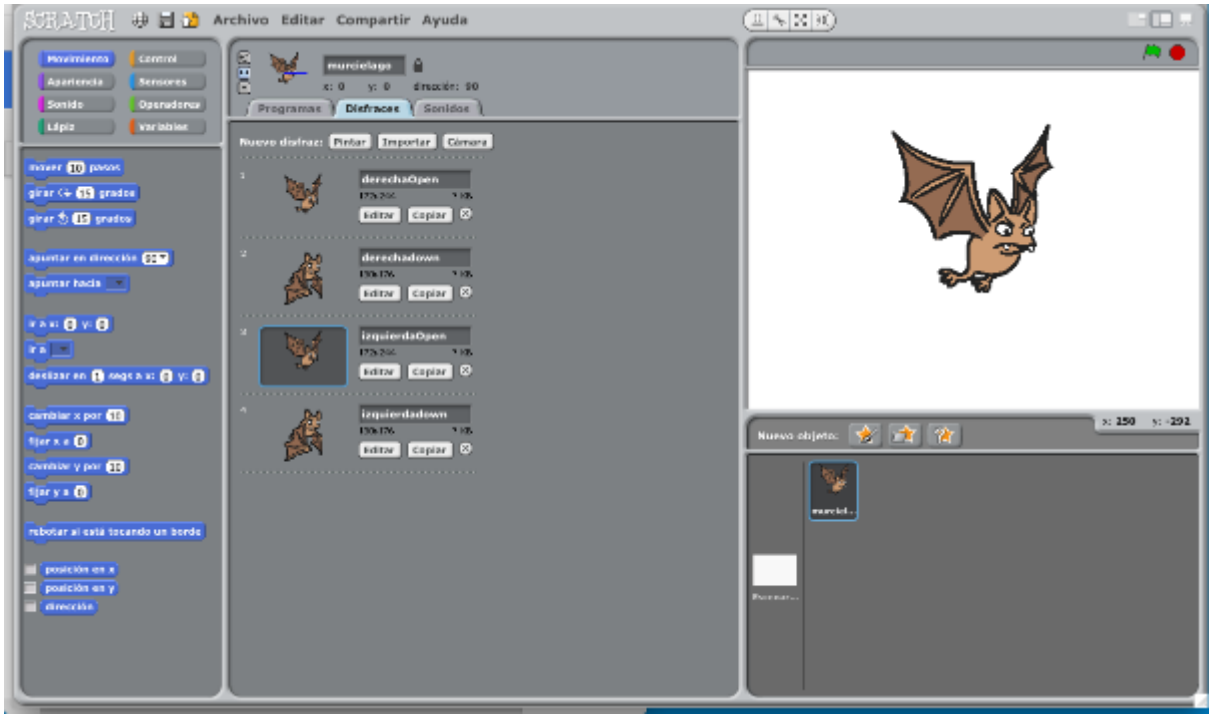


Finalmente hacemos click en aceptar para importar la imagen.

A esta, la nombramos derechaDown



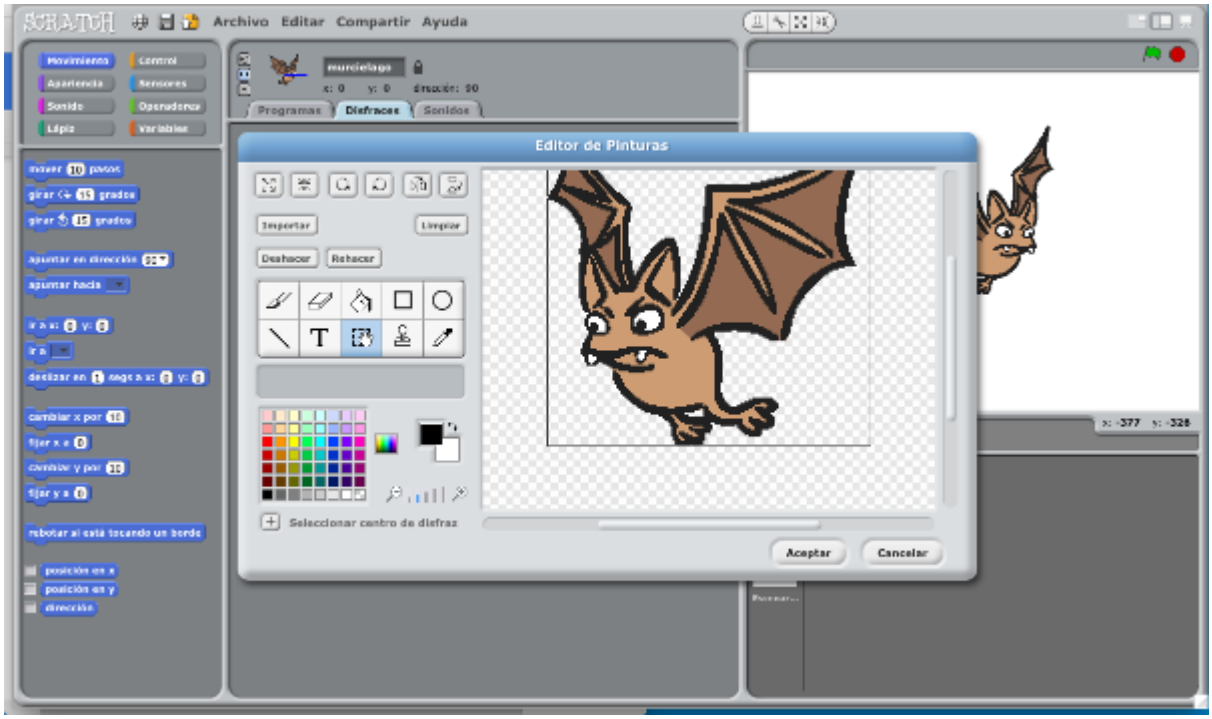
Luego haremos click en el botón copiar de la imagen *derechaOpen* nombrando esta nueva copia como *izquierdaOpen.* Posteriormente, haremos lo mismo con la imagen *derechaDown* nombrando esta nueva copia como *izquierdaDown.* Luego la pantalla tendrá este aspecto.



Para lograr que el personaje se desplace por todo el escenario, los nuevos sprites *izquierdaOpen* e *izquierdaDown* deben ser transformados en su imagen contraria. Para lograr esto, haremos click en el botón editar de la imagen *izquierdaOpen* donde se nos abrirá nuestra ventana de edición de imágenes.

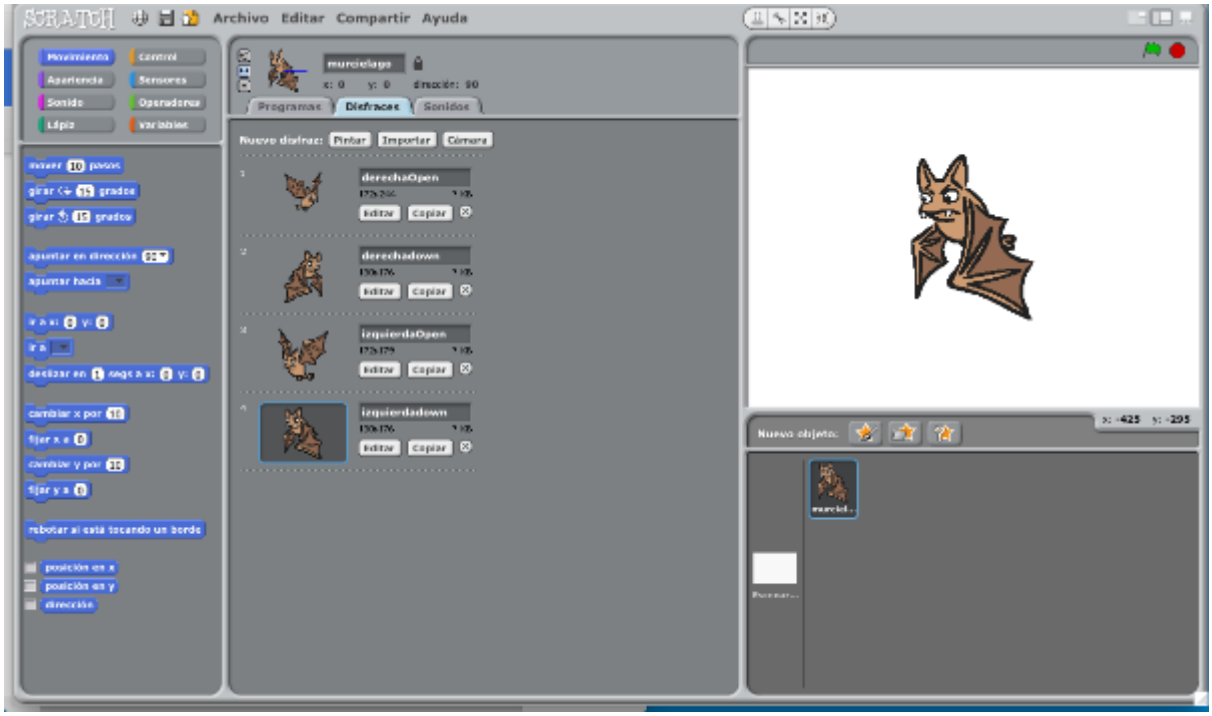
Haremos click en el botón **girar horizontalmente** que tiene el siguiente aspecto.

Finalmente nuestro personaje quedará girado.



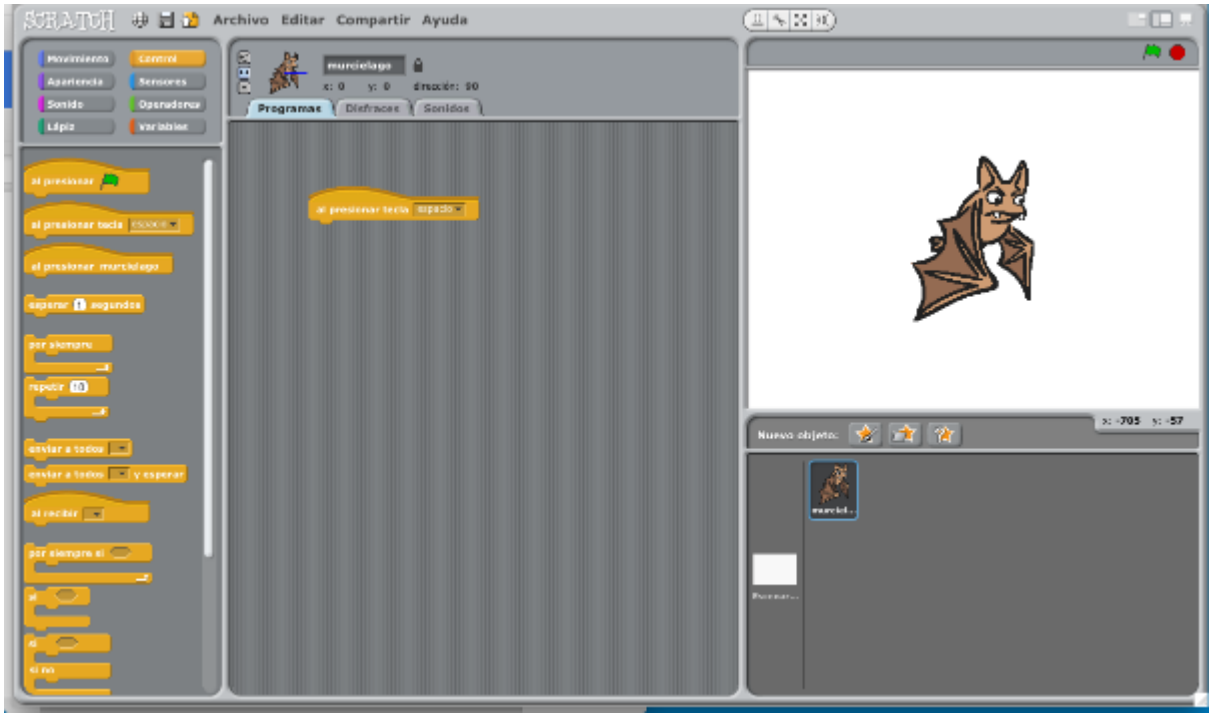
haremos los mismos pasos para el sprite *izquierdaDown.*

Después de finalizar el paso anterior, la pantalla luce de la siguiente forma.



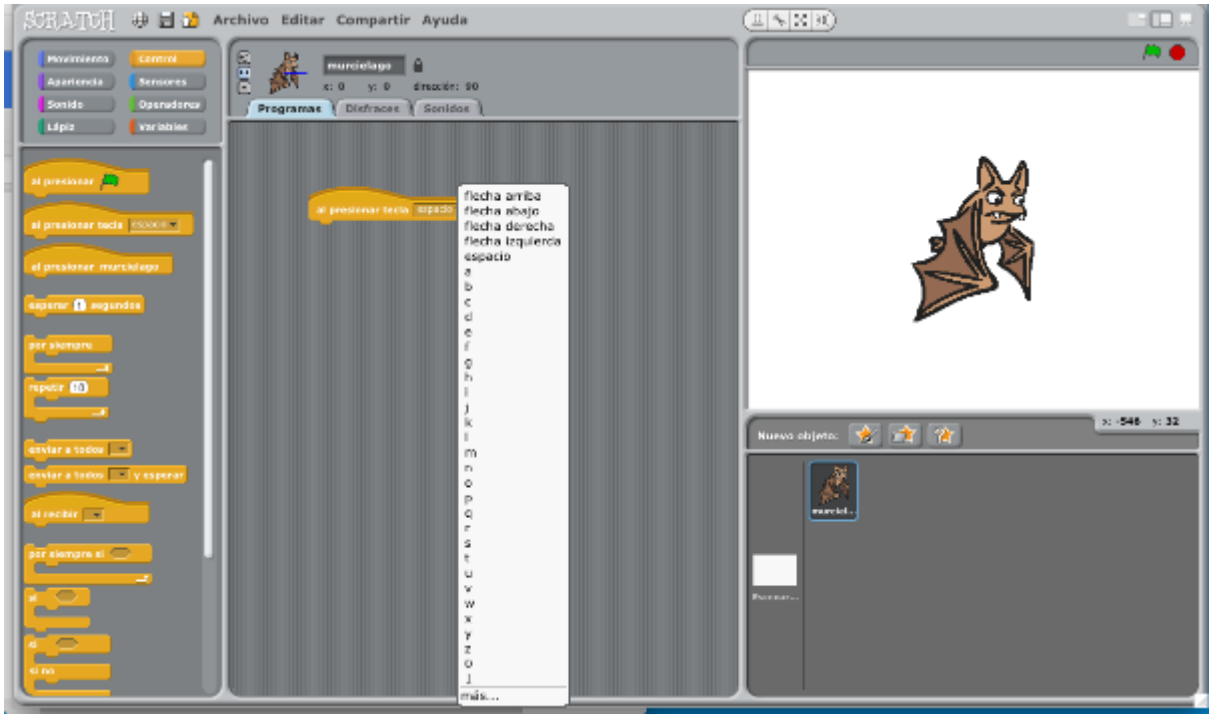
Volveremos a nuestra pantalla de programación haciendo click en la pestaña Programas, que estará totalmente vacía.

En los siguientes pasos empezaremos a programar a nuestro personaje. Para esto, haremos click en el el botón **control** de color amarillo para que nos despliegue las funcionalidades de esta sección. Buscaremos una función que se llama *al presionar tecla "espacio"* ubicada en la parte derecha. Haciendo click sostenido sobre este bloque, lo arrastraremos a la parte central vacía del programa. la pantalla debe de verse de la siguiente manera.



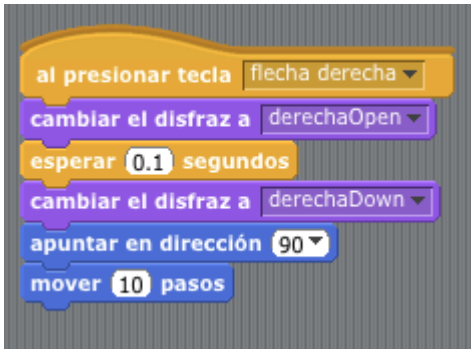
Después de hacer lo anterior. Cambiaremos el parámetro de este bloque, haciendo click en la palabra espacio, encerrada dentro de un rectángulo de color amarillo más oscuro.

Se nos desplegaran varias opciones, y escogeremos la que dice flecha derecha.



De este modo, cada vez que apretemos la tecla derecha, este bloque sabrá exactamente cuándo y que tecla apretamos, para hacer alguna acción en nuestro programa. Que en este tutorial, es hacer que nuestro personaje vuelve hacia la derecha.

Una vez entendido lo anterior, haremos lo siguiente que está en esta imagen, buscando los bloques según su familia de color y en su respectivo orden.

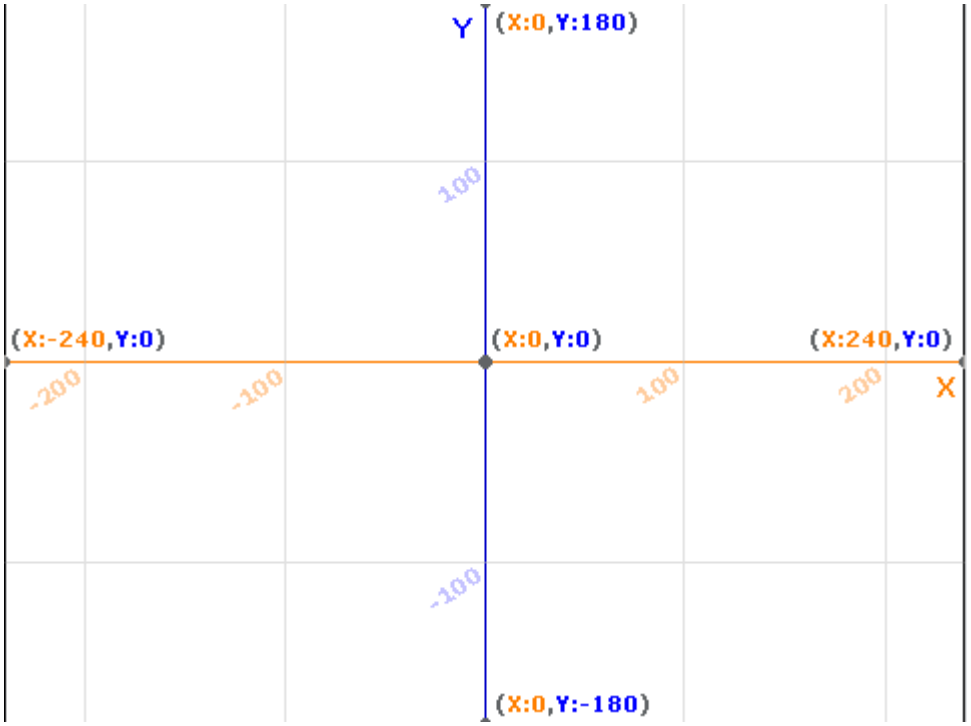


Explicamos este gran bloque color por color. Después del bloque al presionar tecla..., arrastramos un bloque de la sección **apariencia** llamado *cambiar el disfraz a...* Donde tendra un menu para escoger que sprite queremos, eligiendo el sprite que llamamos anteriormente *derechaOpen.*

Después de este cambio, volvemos a la sección **control,** buscando un bloque llamado *esperar 1 segundo,* ese valor en número puede ser cambiado o reemplazado por una variable. En este caso lo cambiaremos a 0.1 segundos para hacer que la animación cumpla con los principios básicos de la animación. En este caso la fluidez.

Después de esto volvemos a la familia **apariencia** y repetimos el paso anterior con el bloque *cambiar disfraz,* reemplazando su valor por *derechaDown.*

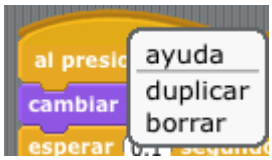
Para entender cómo desplazamos nuestro personaje dentro del escenario. Observaremos las siguiente plano cartesiano.



De esta manera entendemos que el punto (0,0) en nuestro punto central.

Entendido lo anterior, vamos a nuestra familia **Movimiento** y buscamos el bloque que se llama *apuntar en dirección 90.* Dejando su valor en 90 para que el espite mire hacia la derecha.Finalmente agregamos un bloque de la misma familia llamado *mover 10 pasos* para que nuestra imagen se desplaze por el plano 10 pasos hacia la derecha.

Para hacer que nuestro personaje vuelve hacia la izquierda, colocamos nuestro puntero encima del bloque *al presionar tecla flechaDerecha* y haremos click derecho. Aparecerán tres opciones entre ellas *duplicar,* haremos click en ella para duplicar el grupo de bloques.



En este nuevo bloque, cambiaremos el parámetro del bloque *al presionar tecla flechaderecha* por *flechazquierda.* En los bloques de *cambiarDisfraz,* por su respectivo *izquierdaOpen* e *izquierdaDown.* La dirección la dejaremos por defecto y el bloque *mover 10 pasos,* cambiaremos el valor 10 por -10. Para que el murciélago vuelve hacia la izquierda.

Para lograr que nuestro personaje vuele hacia arriba y hacia abajo, duplicamos de nuevo otro bloque. Realizando los mismos pasos anteriores, modificando modificando el valor de cada bloque para lograr la animación deseada.

Invitamos a que el lector experimente un poco para lograr lo anterior.

Para finalizar, cambiaremos el fondo por una ciudad, seleccionando el *escenario* y eligiendo un fondo de nuestra elección.

Puedes descargar el programa haciendo click aqui → archivo animacionSimple

Dibujando una casa a través de condiciones simples.

Normalmente cuando estamos programando animaciones y juegos, necesitamos que pasen determinados acontecimientos o acciones según condiciones o sucesos a medida que transcurre el programa. Normalmente es hacer determinada cosa según si una condición es verdadera o falsa.

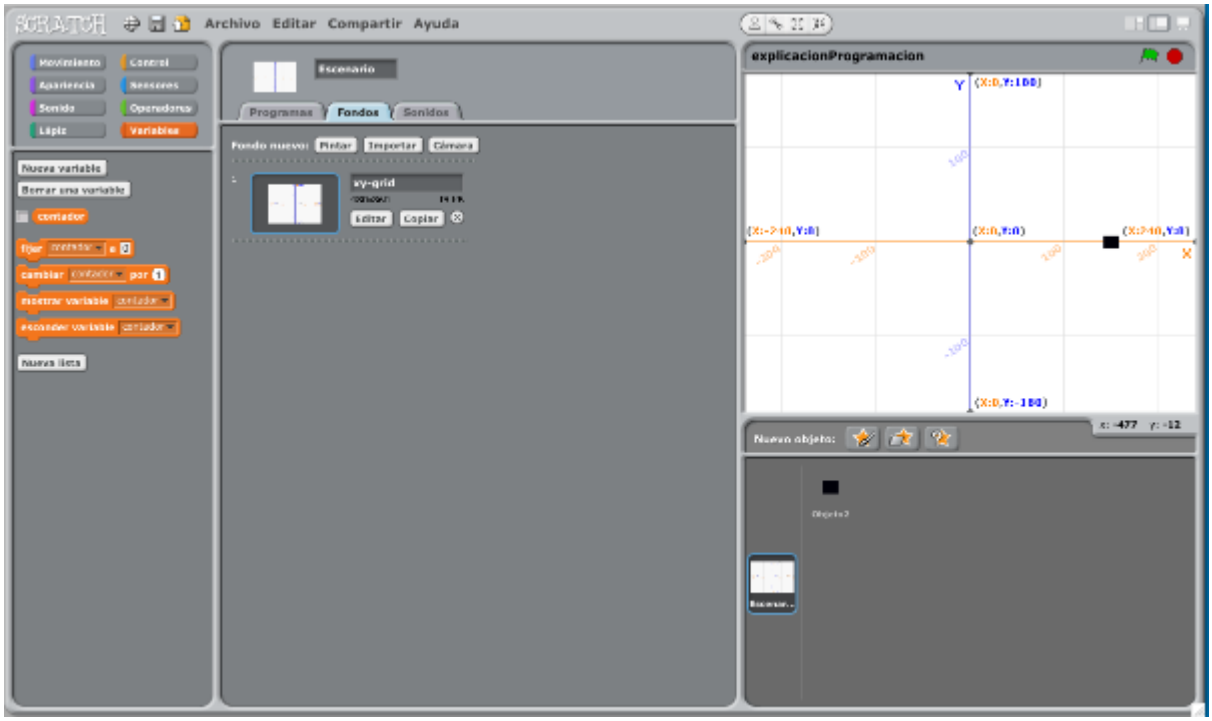
Es muy normal en los juegos por ejemplo perder o ganar vidas, estos factores de cambio en esos valores pueden ser sucesos que ocurren a través de un condicional y su única condición falsa o verdadera.

Si **un enemigo choca con un disparo del player** *entonces* **desaparece de la pantalla y sumamos 1 al score (puntaje).**

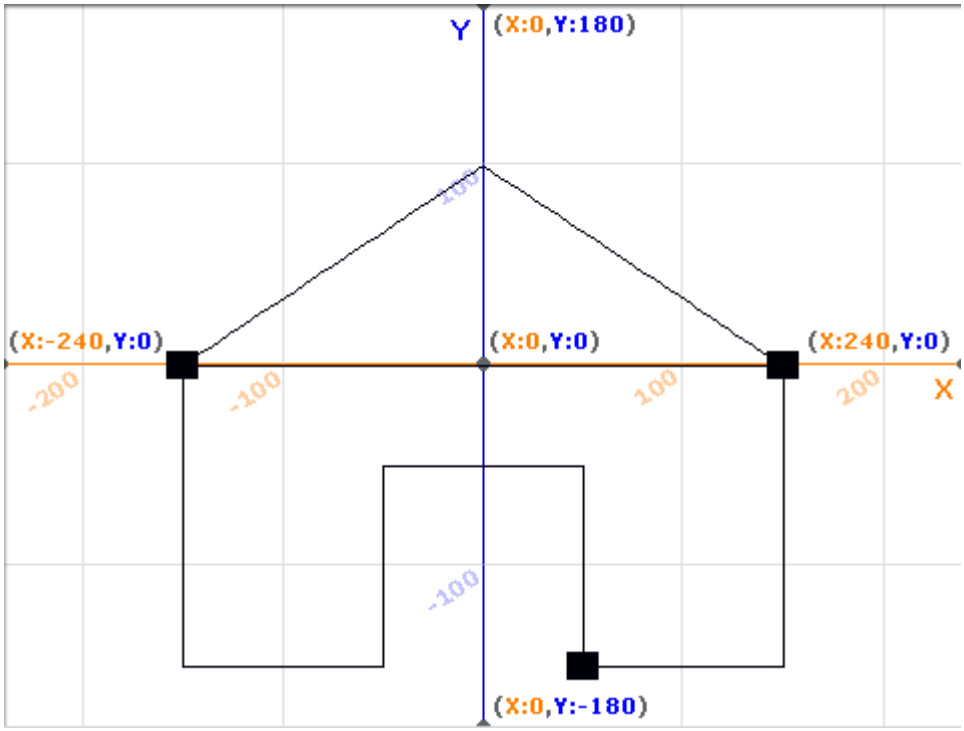
Si nos detenemos a leer nuevamente el párrafo anterior notamos que hay dos palabras claves que hacen que un acontecimiento ocurra después del otro, las palabras *si* y *entonces*.

Objetivo: En este ejercicio dibujaremos una casa a partir de condicionales y un contador que desencadenara cada acción a ser dibujada en ese momento específico.

Para empezar recordamos que nuestro punto central está en la mitad del lienzo. Empezamos nuestro programa con seleccionar el escenario y luego importamos la imagen de fondo que contiene las coordenadas cartesianas.



De esta manera, si queremos dibujar una casa. Tendremos que decirle a nuestro punto las coordenadas exactas a seguir. Es como dibujar una figura sin levantar el lápiz del papel, en este caso dibujaremos una simple casa.



Como podemos observar en la anterior imagen. Disponemos de ciertas coordenadas claves en el dibujo, que sobresalen como puntos negros en el sketch. Para hacer el anterior programa, crearemos una variable llamada contador inicializandola a cero.

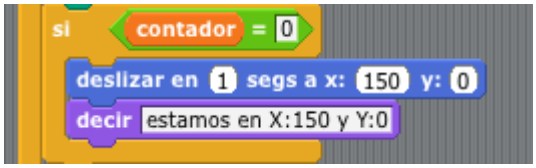
Ella será la encargada de estar pendiente del evento presionar tecla. Cada vez que apretemos la tecla espacio de nuestro computador, se le sumara 1 a esta variable, causando que aumente y cambie el valor del contador.

si dibujamos nuestra casa en el lienzo. Tendremos que tener en cuenta las coordenadas en el lienzo y como scratch maneja el desplazamiento.

Los movimientos en scratch son relativamente sencillos según el problema, en este caso basta entender que para desplazarnos a la izquierda solo tenemos que escribir números negativos en las coordenadas x de los bloques ir y desplazar.

Inversamente, para desplazarnos a la derecha escribimos números positivos.

Para la coordenada en Y, basta entender que si queremos movernos hacia abajo, escribimos números positivos, y si queremos desplazarnos hacia arriba entonces escribimos números negativos.



En este link puedes descargar el programa completo