

## Scratch



### 1. ¿Que es scratch?

Scratch es un programa desarrollado por el MIT, diseñado para enseñar programación sin necesidad de tener conocimientos previos en los lenguajes de programación. Existen varias versiones diferentes aunque el funcionamiento es similar.

**Como características principales podemos destacar:**

- Es gratis. No es necesario pagar ningún tipo de licencia.
- Existen versiones para distintos sistemas operativos y distintos dispositivos
- Existen versión online.

En la página <https://scratch.mit.edu/> puedes crear programas sin ni siquiera darte de alta como usuario, aunque también puedes hacerlo.

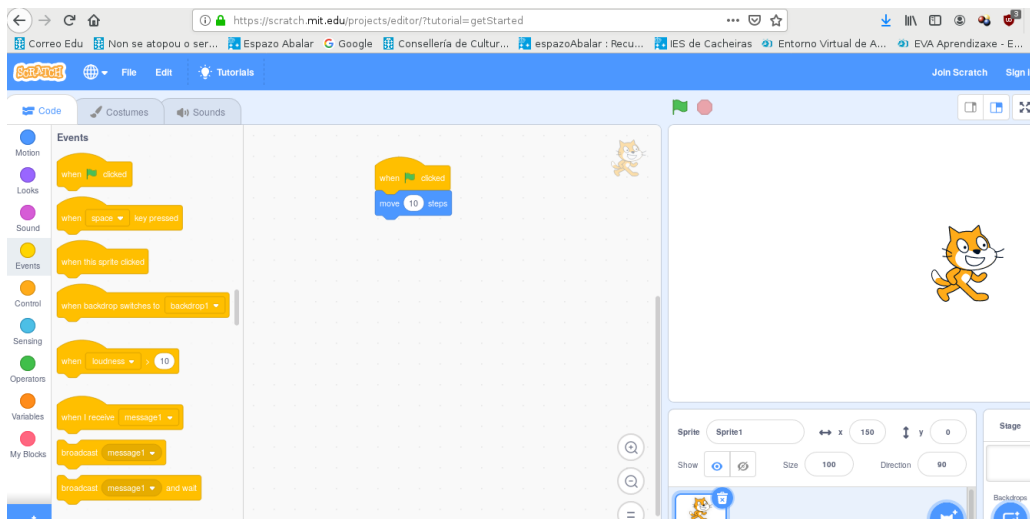


Figura 1: Interface de scratch.mit.edu

- Puedes guardar tus proyectos en tu ordenador o subirlos a la nube y compartirlos con toda la comunidad scratch.
- Puedes ver y valorar los proyectos que diariamente van subiendo y compartiendo millones de usuarios.
- Puedes descargar e instalar el programa en tu dispositivo y trabajar sin conexión sobre tus proyectos en cualquier lugar

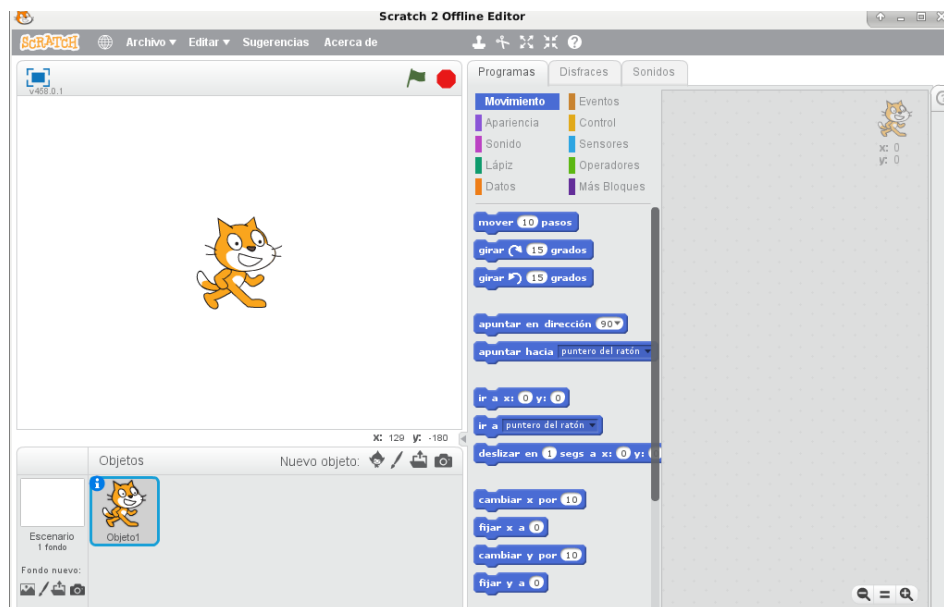


Figura 2: Interface de Scratch 2

En lugar de utilizar las habituales instrucciones de código para crear el algoritmo, con Scratch utilizamos **bloques**, una especie de piezas de puzzle que encajan o no entre ellas dando una visión intuitiva de lo que estamos haciendo. De esta forma el algoritmo se puede leer e interpretar fácilmente sin necesidad de tener conocimientos de programación.



Figura 3: Bloques de programación scratch

```
def add5(x):
    return x+5

def dotwrite(ast):
    node_name = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '%s (label="%s" % (node_name,label),
    if isinstance(ast[1], str):
        print = "%s";' % ast[1]
    else:
        print '"'
    else:
        print '"';'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % node_name,
        for name in children:
            print '%s' % name,
```



*Figura 4: Lenguaje de código.  
Python*

## 2.- La interface de scratch



Figura 5: Interfaz de scratch

En la interfaz de scratch nos encontramos varias zonas, cada una de ellas tiene su función

### 1.- Escenario



Figura 6: Escenario de scratch con fondo

El escenario tiene 480 puntos (píxeles) de ancho y 360 puntos de alto y está dividido en un plano cartesiano  $x-y$ . El centro del escenario corresponde a las coordenadas  $X:0, Y:0$ .

Podemos encontrar las coordenadas del escenario moviendo el ratón por el mismo y viendo la información de posición en la parte inferior derecha

X: 240 Y: -180

## 2.- Lista de objetos

La lista de Objetos muestra imágenes en miniatura de todos los Objetos disponibles en el proyecto. Cada Objeto, muestra su nombre bajo la imagen.

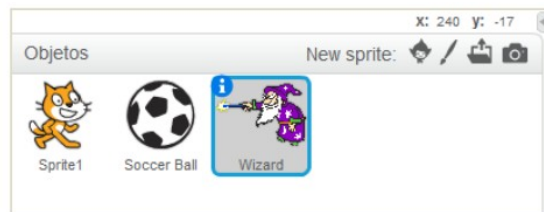


Figura 7: Lista de objetos de Scratch

Podemos ver y editar los programas, los disfraces y los sonidos de los Objetos haciendo clic sobre la imagen miniatura del Objeto deseado, en la Lista de Objetos o haciendo doble clic sobre el Objeto en el Escenario, en este caso, el Objeto seleccionado se resalta y aparece enmarcado en azul, en la Lista de Objetos).

Para mostrar/esconder, exportar, duplicar o borrar un Objeto, tenemos que hacer clic derecho sobre su imagen miniatura en la Lista de Objetos.

Podemos modificar la apariencia de un Objeto cambiando de disfraz.

Podemos utilizar una serie de objetos prediseñados que aparecen en el programa scratch o podemos crearlos nosotros en un editor de dibujo del propio programa.

### 3.- Escenario

Los objetos de scratch aparecen y se mueven con un escenario (stage) de fondo, el cual tiene unas dimensiones de 480x360 pixeles.

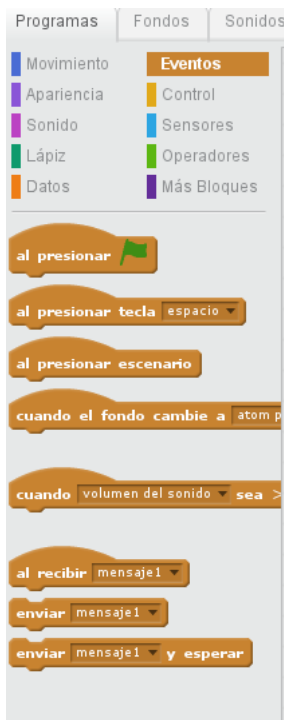
El Escenario puede cambiar su apariencia intercambiando fondos.

Al igual que en el caso de los objetos podemos escoger un fondo de los que aparecen por defecto en scratch, subir uno de un archivo de imagen de nuestra propia cámara o bien descargar una imagen de internet.



*Figura 8: Escenario de Scratch*

### 3.- Paleta de bloques



En la parte central de la interfaz, en el desplegable "programas" se encuentra la paleta de bloques, donde podemos encontrar todas las instrucciones para construir nuestros programas.

Encontramos en este desplegable las opciones:

- **Movimiento:** Principalmente instrucciones para mover un objeto dentro del escenario
- **Apariencia:** Instrucciones encaminadas a cambiar los disfraces de los distintos personajes, los fondos, generar diálogos...
- **Sonido:** Podemos crear, importar o grabar sonidos
- **Lápiz:** Podemos desarrollar programas creadores de dibujos, con esta herramienta podemos dibujar sobre el escenario.
- **Datos:** Podemos crear **variables** que se modifiquen según el desarrollo del programa como por ejemplo un score o puntuación.
- **Eventos:** Sirve para enlazar un suceso con una acción, por ejemplo, al presionar una tecla.... la acción será otra parte del programa.

Figura 9: Paleta de bloques de Scratch

- **Control:** Es una de las partes más importantes del programa, permite crear condiciones para la ejecución del programa, en este punto se encuentran muchas de las órdenes más habituales en un lenguaje de programación.

- **Operadores:** En esta categoría encontramos distintos bloques que nos servirán para **realizar cálculos numéricos, entre variables o valores de bloques** que ya existen.
- **Sensores:** Estos bloques sirven para crear condiciones relacionadas con las interacciones entre elementos como el puntero ratón, el un borde de la pantalla, un color concretos o con otros personajes.
- **Más bloques:** Este bloque permite que nosotros mismos como programadores creamos funciones nuevas para el programa.

#### 4.- Área de programa



Figura 10: Área de programa de Scratch

El área de programa es el lugar donde escribimos nuestro algoritmo o programa, una secuencia de bloques que arrastramos desde la paleta de bloques hasta aquí.

Es posible realizar ciertas acciones sobre los bloques o los conjuntos de bloques, como duplicarlos o eliminarlos.

El programa scratch también permite modificar fácilmente el algoritmo.

## 6.- Los objetos

Los objetos de Scratch son entidades gráficas, con una forma, conjunto de colores y tamaños determinados, también recibe el nombre de **sprites** ...

Podemos incorporar **objetos** en un proyecto de **Scratch** de diferentes formas:



- A partir de la biblioteca de **objetos** que tiene **Scratch**.

**Scratch** incorpora un conjunto de imágenes que pueden ser utilizadas como **objetos** dentro de tu proyecto.

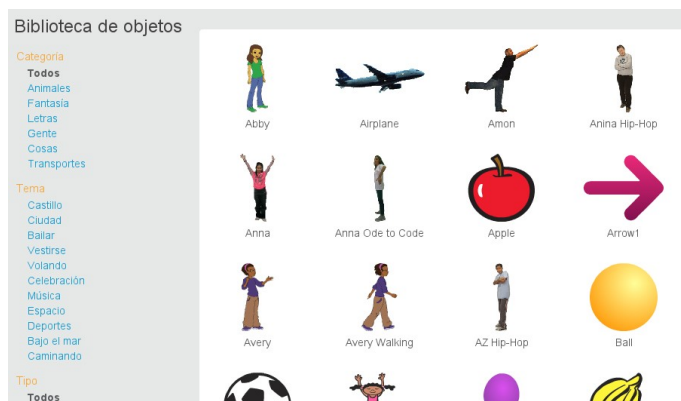


Figura 11: Biblioteca de objetos de Scratch

- Podemos crear o editar un objeto con el editor de objetos

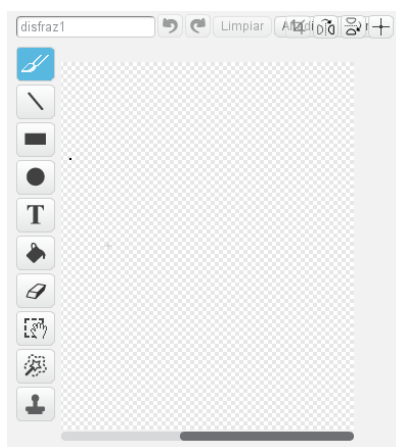


Figura 12: Editor de objetos de Scratch

- **Cargar un objeto desde un archivo:**



Importar una imagen o un objeto guardado en alguna carpeta del disco duro del computador. Si lo que queremos es cargar una imagen conviene tener presente las transparencias y los derechos de autor.

- **Crear un nuevo objeto utilizando la cámara:**



La aplicación pide permiso para acceder a la webcam del equipo informático para capturar imágenes.

**En la Comunidad de Scratch existen varios estudios con multitud de proyectos que hacen las veces de bibliotecas de objetos o *sprite packs*. Su único propósito es almacenar objetos, disfraces, escenarios, etc. Uno de este tipo de estudios, de los más completos, es el siguiente: <https://scratch.mit.edu/studios/7155/>**

## 7.- Disfraces y sonidos

Por defecto cada objeto lleva asociados uno o varios disfraces y uno o varios sonidos en ambos casos son editables, es decir, podemos realizar modificaciones sobre ellos.

Los disfraces o "costumes" en Scratch son las distintas apariencias que puede mostrar un objeto en la pantalla, sirven para dar dinamismo o mejorar los movimientos. Hay objetos que pueden requerir varios disfraces, por ejemplo un personaje que requiera movimientos en las piernas para simular que anda. En otros objetos puede ser suficiente con un disfraz, por ejemplo una pelota monocolor.

Los disfraces pueden ser editados de una forma similar a los objetos

Los sonidos

## 8.- Fondos

El fondo (backdrop) es la imagen que ocupa el escenario de scratch, esta imagen está sujeta a la programación y podemos hacer que varíe con el tiempo o cuando se produzca determinado suceso.

Podemos escoger un fondo a partir de una biblioteca de fondos, editarlo nosotros, descargar e importar una imagen obtenida de internet, en este caso es importante que esté libre de derechos). También podemos hacer una fotografía con la webcam y usarla como fondo



## 9. Programando

Ahora vamos a empezar a programar, es importante entender algunos conceptos de programación que nos pueden ayudar:

### 9.1 Secuencia

Una secuencia es una sucesión de pasos ordenados. Se indica en que orden deben ocurrir los distintos sucesos o acciones del programa

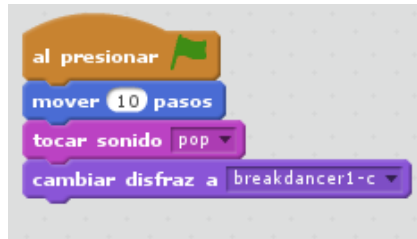


Figura 13: Ejemplo de secuencia

### 9.2 Iteración

Indica que, según cierta condición, un número de instrucciones podrían repetirse un numero determinado o incluso indeterminado de veces. Las iteraciones son básicamente estructuras ciclicas que nos permitirán repetir una cantidad de veces determinada o indeterminada unas instrucciones.

Los bloques "por siempre" y "repetir" son bloques típicos de iteración.



Figura 14: Ejemplo de iteración

### 9.3 Decisiones o condiciones

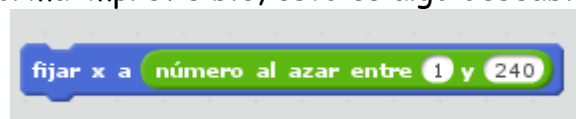
Las decisión o condiciones indican que según unas ciertas condiciones que establecemos en la programación se ejecutarán o no un conjunto de instrucciones. Los bloques si o si no son ejemplos de condiciones de programación



Figura 15: Ejemplo de condición

### 9.4 Números aleatorios

Los números aleatorios permiten que el azar participe en la programación o que el programa actúe de forma imprevisible, esto es algo deseable en general los juegos.



## 10 Trabajando con los bloques

Los bloques son las unidades básicas de programación de scratch, en los distintos submenús nos podemos encontrar distintos tipos de bloques, su color y su forma nos ofrecen información acerca de su uso

### Bloques de inicio



Sirven para iniciar y no pueden ir a continuación de otro bloque,

### Bloques de desarrollo:



Incluyen una secuencia del algoritmo, pueden ir a continuación de otro bloque

### Bloques numéricos:



Son bloques con bordes redondeados. Tienen que ir dentro de otro bloque, solo permiten valores numéricos

### Bloques alfanuméricos:



Son bloques con forma de cuadrado o rectángulo. Pueden incluir caracteres numéricos y no numéricos, letras, símbolos...

### Bloques lógicos:



Son bloques con forma de polígono de 6 lados. Incluyen operaciones lógicas de resultado verdadero o falso

## 10.1 Movimiento

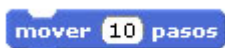
En el submenú movimiento nos encontramos los bloques necesarios para mover y orientar los objetos. Se utilizan sobretodo para situar y mover los objetos o personajes dentro del escenario.

Comportamiento de los bloques de movimiento

| MOVIMIENTO |                                                                                                 |
|------------|-------------------------------------------------------------------------------------------------|
|            | Mueve el Objeto hacia adelante o hacia atrás. Con valores + ó -.                                |
|            | Rota el Objeto en el sentido de las manecillas del reloj. Con valores + ó -.                    |
|            | Rota el Objeto en el sentido contrario a las manecillas del reloj.                              |
|            | Apunta el Objeto en la dirección especificada (0=arriba; 90=derecha; 180=abajo; -90=izquierda). |
|            | Apunta el Objeto hacia el puntero del ratón o hacia otro Objeto.                                |
|            | Mueve el Objeto hacia una posición específica de X, Y en el escenario.                          |
|            | Mueve el Objeto a la ubicación del puntero del ratón o de otro Objeto.                          |
|            | Mueve el Objeto suavemente a una posición determinada en un lapso de tiempo específico.         |
|            | Cambia la posición X del Objeto en una cantidad determinada (incrementa).                       |
|            | Fija la posición X del Objeto a un valor específico.                                            |
|            | Modifica la posición Y del Objeto en una cantidad determinada (incrementa).                     |
|            | Fija la posición Y del Objeto a un valor específico.                                            |
|            | Gira el Objeto en sentido contrario, cuando este toca un borde del escenario.                   |
|            | Informa la posición X del Objeto. (Rango entre -240 a 240)                                      |
|            | Informa la posición Y del Objeto (Rango entre -180 a 180)                                       |
|            | Informa la dirección del Objeto (0=arriba; 90=derecha; -90=izquierda; 180=abajo)                |

Figura 16: Bloques de movimiento de Scratch

### 10.1.1 El bloque mover



El bloque mover desplaza el objeto un número determinado de pasos en la dirección señalada por dicho objeto. Cada paso se corresponde con un pixel en el escenario. Si ponemos un valor negativo, el objeto se desplazará en la dirección opuesta a la señalada.

#### Ejemplo

Puedes comprobar el funcionamiento introduciendo las siguientes instrucciones en el programa



### 10.1.2 Los bloques girar



Los bloques girar hacen que el objeto gire en el sentido de la flecha introducido en el mismo bloque el número de grados sexagesimales que pongamos.

#### Ejemplos

Puedes comprobar el funcionamiento introduciendo las siguientes instrucciones en el programa



Ahora observa lo que pasa introduciendo un bloque de control, repetir y combinando mover y girar.



### 10.1.2 Los bloques apuntar

Los bloques apuntar hacen que el objeto cambie la dirección de su movimiento.



Con el bloque apuntar en dirección, asignamos nosotros la dirección en grados sexagesimales con un valor numérico.



Con el bloque apuntar hacia, hacemos que el objeto se dirija hacia otro objeto o hacia el puntero del ratón.

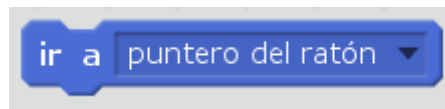
Puedes probar como funciona con el siguiente algoritmo



### 10.1.3 Bloques ir a y deslizar



El bloque ir a x:\_\_\_ y:\_\_\_ dirige el objeto instantáneamente a las coordenadas especificadas, mientras que el bloque deslizar en\_\_\_ a x:\_\_\_ y:\_\_\_ lo dirige en el tiempo indicado.



El bloque ir a (elige objeto) tiene también un efecto instantáneo dirigiendo el objeto hacia el propósito elegido.

El bloque deslizar hace que el objeto se desplace durante un tiempo determinado hacia unas coordenadas concretas.



### 10.1.4 Bloques cambiar x por y cambiar y por

Estos bloques producen cambios de posición en los ejes determinados en un numero de pixeles.

### 10.1.5 Bloques fijar x y fijar y

Estos bloques hacen que un objeto solo pueda desplazarse en un eje horizontal o vertical definido por la x o y correspondiente.

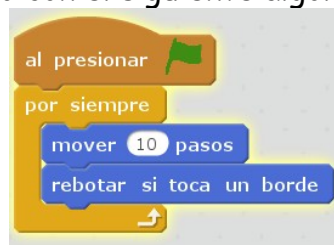
Puedes probar como funciona el bloque fijar con el siguiente algoritmo.



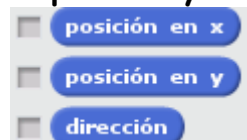
#### 10.1.6 El bloque rebotar

Este bloque comprueba si un borde del panel de acción es tocado. Si es el caso, el objeto rebotará.

Puedes probar el funcionamiento con el siguiente algoritmo



#### 10.1.7 Los bloques de variables de posición y dirección



Estos bloques de forma redondeada encajan en algunos huecos de otros bloques, por ejemplo y especialmente en los de operadores.

Puedes probar como funciona con el siguiente algoritmo.



## 10.2 Los bloques de apariencia

### Comportamiento de los bloques de apariencia

| APARIENCIA                                                                  |                                                                                                                                                                       |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cambiar el disfraz a</b> <code>costume2</code>                           | Modifica la apariencia del Objeto cambiando de disfraz.                                                                                                               |
| <b>siguiente disfraz</b>                                                    | Cambia el disfraz del Objeto por el siguiente disfraz en la lista de disfraces (cuando llega al final del listado de estos, vuelve a comenzar con el primer disfraz). |
| <b># de disfraz</b>                                                         | Informa el número correspondiente al presente disfraz del Objeto                                                                                                      |
| <b>cambiar el fondo a</b> <code>Background1</code>                          | Modifica la apariencia del escenario cambiando a un fondo diferente.                                                                                                  |
| <b>fondo siguiente</b>                                                      | Modifica la apariencia del escenario pasando al siguiente fondo disponible en el listado de estos.                                                                    |
| <b># de fondo</b>                                                           | Reporta el número del fondo actual del escenario.                                                                                                                     |
| <b>decir</b> <code>¡Hola!</code> <b>por</b> <code>2</code> <b>segundos</b>  | Despliega una nube de diálogo del Objeto durante un lapso de tiempo determinado                                                                                       |
| <b>decir</b> <code>¡Hola!</code>                                            | Despliega una nube de diálogo del Objeto (se puede eliminar esta burbuja de diálogo ejecutando este bloque sin texto alguno).                                         |
| <b>pensar</b> <code>mmm...</code> <b>por</b> <code>2</code> <b>segundos</b> | Despliega una nube de pensamiento del Objeto durante un determinado lapso de tiempo.                                                                                  |
| <b>pensar</b> <code>mmm...</code>                                           | Despliega una nube de pensamiento del Objeto.                                                                                                                         |
| <b>cambiar efecto</b> <code>color</code> <b>por</b> <code>25</code>         | Modifica un efecto visual del Objeto en una cantidad especificada (use el menú desplegable para seleccionar el efecto).                                               |
| <b>fijar efecto</b> <code>color</code> <b>a</b> <code>0</code>              | Establece un efecto visual a un número dado (la mayoría de efectos visuales se ubica en un rango de 0 a 100).                                                         |
| <b>quitar efectos gráficos</b>                                              | Limpia o borra todos los efectos gráficos de un Objeto                                                                                                                |
| <b>cambiar tamaño por</b> <code>10</code>                                   | Modifica el tamaño del Objeto en una cantidad especificada (incrementa).                                                                                              |
| <b>fijar tamaño a</b> <code>100 %</code>                                    | Ajusta el tamaño del Objeto en un porcentaje (%) específico respecto a su tamaño original.                                                                            |
| <b>tamaño</b>                                                               | Informa el tamaño del Objeto como porcentaje (%) de su tamaño original.                                                                                               |
| <b>mostrar</b>                                                              | Hace aparecer un Objeto en el escenario.                                                                                                                              |
| <b>esconder</b>                                                             | Hace desaparecer un Objeto del escenario (cuando el Objeto está escondido, otros Objetos no lo pueden detectar con el bloque "¿tocando?").                            |
| <b>enviar al frente</b>                                                     | Ubica el Objeto al frente de todos los demás Objetos (capa superior).                                                                                                 |
| <b>enviar hacia atrás</b> <code>1</code> <b>capas</b>                       | Mueve el Objeto hacia atrás, un número determinado de capas, de manera que pueda ocultarse detrás de otros Objetos.                                                   |

### 10.2.1 Bloques decir y pensar

Con estos bloques aparece sobre el objeto un bocadillo de cómic con las palabras que queramos



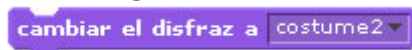
### 10.2.2 Bloques mostrar y ocultar

Los bloques mostrar y ocultar sirven para que en determinadas condiciones los objetos aparezcan o desaparezcan.

Puedes probar como funciona con el siguiente algoritmo

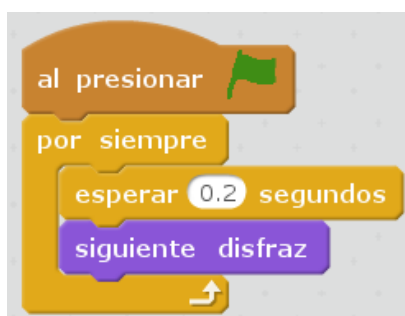


### 10.2.3 Bloques cambiar disfraz a siguiente disfraz



El bloque cambiar disfraz permite cambiar la apariencia del objeto al disfraz que escojamos, mientras que siguiente disfraz hace que cambie la apariencia por el siguiente de la lista.

Para probar estos bloques vamos a escojer como objeto al personaje AZ Hip Hop

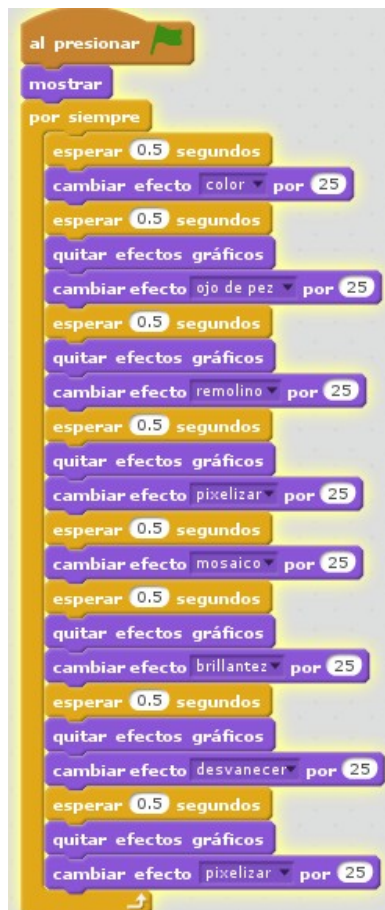


### 10.2.4 Bloques de efectos

Los bloques de efectos permiten aplicar una serie de efectos a nuestro objeto:

- Color: Modifica el color del disfraz.
- Ojo de pescado: Genera el efecto de un cristal ojo de pez.
- Remolino: distorsiona la imagen generando efectos concéntricos.
- Pixelar: convierte el objeto en píxeles de diferentes formas y colores.
- Mosaico: repite el objeto la cantidad de veces que le indiquemos en forma de imágenes sucesivas cuadradas.
- Brillantez: cambia el nivel de brillo del objeto.
- Fantasma: a medida que aumentamos los valores se vuelve transparente.

Puedes comprobar lo que pasa con el siguiente algoritmo y probar con las variaciones que quieras.



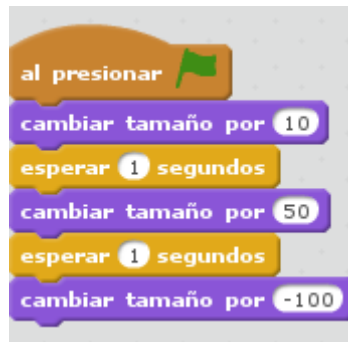
### 10.2.5 Bloques de tamaño



Los bloques de tamaño permiten modificar el tamaño del objeto.

El bloque cambiar tamaño por... con valores positivos aumenta el tamaño del objeto y con valores negativos los disminuye.

Fijar tamaño proporciona un valor que es un porcentaje respecto al tamaño inicial definido del objeto.



La variable tamaño permite conocer el tamaño en un momento dado y si este varía, utilizarlo en el algoritmo.

## 10.3 Los bloques de sonido

En el submenú SONIDOS podemos encontrar numerosos bloques relacionados con los efectos de sonido que podemos asignar a nuestros programas.

Estos bloques nos van a permitir asignar sonidos a nuestros objetos, poner música o efectos de fondo, modificar el ritmo, cambiar el volumen, etc.

| SONIDOS                                                                                            |                                                                                          |
|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>tocar sonido</b> <span>miau</span>                                                              | Toca el sonido indicado, la ejecución del programa continúa.                             |
| <b>tocar sonido</b> <span>miau</span> <b>y esperar</b>                                             | Toca el sonido indicado, la ejecución del programa se para hasta que finalice el sonido. |
| <b>detener todos los sonidos</b>                                                                   | Detiene la reproducción de todos los sonidos.                                            |
| <b>tocar tambor</b> <span>48</span> <b>durante</b> <span>0.2</span> <b>pulsos</b>                  | Toca el instrumento seleccionado durante los pulsos indicados.                           |
| <b>silencio por</b> <span>0.2</span> <b>pulsos</b>                                                 | Desactiva el sonido durante los pulsos indicados.                                        |
| <b>tocar nota</b> <span>60</span> <b>durante</b> <span>0.5</span> <b>pulsos</b>                    | Toca la nota seleccionada durante los pulsos indicados.                                  |
| <b>fijar instrumento a</b> <span>1</span>                                                          | Fija el instrumento con el que se tocaran las notas.                                     |
| <b>cambiar volumen por</b> <span>-10</span>                                                        | Aumenta o disminuye el volumen de reproducción de los sonidos.                           |
| <b>fijar volumen a</b> <span>100</span>                                                            | Fija en porcentaje el volumen total de reproducción de los sonidos.                      |
|  <b>volumen</b> | Informa del volumen actual.                                                              |
| <b>cambiar tempo por</b> <span>20</span>                                                           | Aumenta o disminuye el tempo (pulsos por minuto).                                        |
| <b>fijar tempo a</b> <span>60</span> <b>ppm</b>                                                    | Fija el tempo a un número determinado de pulsos por minuto.                              |
|  <b>tempo</b>   | Informa del tempo actual (pulsos por minuto)                                             |

### 10.3.1 Bloque tocar sonido



El bloque "tocar sonido" reproduce el sonido indicado. Una vez se ha comenzado a reproducir el sonido, la ejecución del programa continúa y sigue a la siguiente instrucción.

Los sonidos que podemos elegir son aquellos que se encuentren en la lista de sonidos del objeto activo.

Podemos añadir sonidos al objeto, para ello tenemos 2 opciones; grabarlos  o descargarlos  desde nuestro ordenador.

### 10.3.2 Bloque tocar sonido y esperar

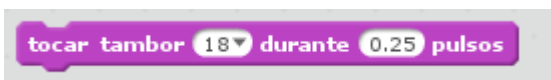
El funcionamiento es muy parecido a tocar sonido, la diferencia está en que en este caso no empieza la siguiente acción hasta que no se acabe el sonido. Puedes comprobar la diferencia con el siguiente algoritmo



### 10.3.3 Bloque detener sonidos

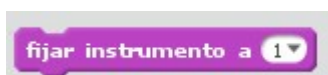
El bloque detener sonidos detiene todos los sonidos al ocurrir una acción

### 10.3.4 Bloque tocar tambor

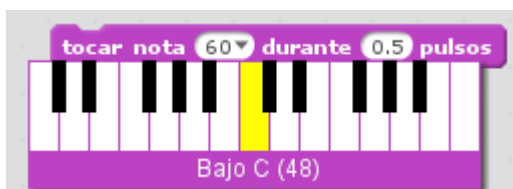


Este bloque cuenta con una amplia lista de instrumentos de percusión que podemos hacer sonar a nuestro gusto, la unidad que podemos utilizar es el pulso, por defecto un pulso equivale a un segundo, pero podemos modificar la duración de un pulso.

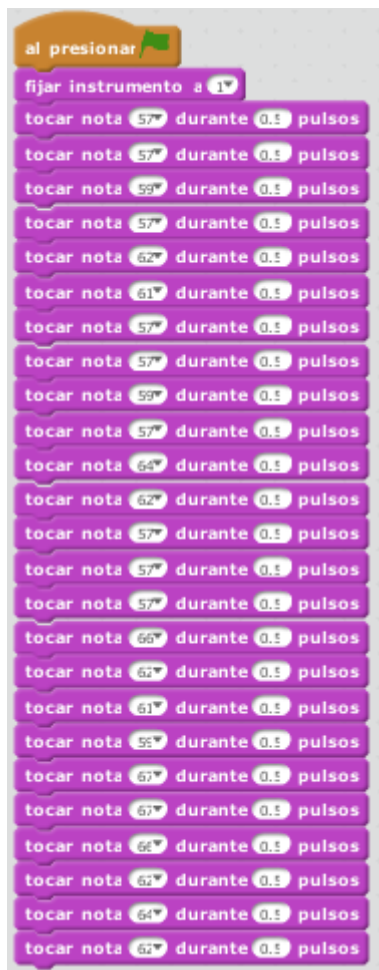
### 10.3.4 Bloque fijar instrumento a ...



Este bloque cuenta con una amplia lista con diversos instrumentos que podemos hacer sonar a nuestro gusto, con el bloque tocar nota podemos construir melodías. Al hacer uso de este bloque nos aparece un piano donde podemos localizar cada una de las notas. En la aplicación aparecen con la nomenclatura anglosajona (A, B, C, D,... que lógicamente tiene su equivalencia con nuestro sistema.



*Do Re Mi Fa Sol La Si Do*  
**C D E F G A B C**



### 10.3.5. Bloques de cambiar volumen, fijar volumen y variable volumen

cambiar volumen por -10

fijar volumen a 100

volumen

Podemos cambiar el volumen en el programa, fijarlo a uno determinado e incluso utilizar el volumen como variable.

### 10.3.6. Bloques de cambiar tempo, fijar tempo y variable tempo

La variable tempo se utiliza para fijar el número de pulsos por minuto. El tempo se utiliza en los bloques "tocar tambor", "silencio" y "tocar nota".

Por defecto el valor de tempo es de 60, por lo que un pulso equivale a un segundo. Cuanto mayor sea el tempo más rápido tocará Scratch los sonidos.

## 10.4 Los bloques de lápiz

La herramienta Lápiz encontramos todos los bloques relacionados con las opciones de dibujo que podemos asignar al movimiento de los objetos incluidos en nuestros programas.

| LÁPIZ                                  |                                                                                                                             |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>borrar</b>                          | Borra todos los trazos generados con el lápiz en todos los objetos.                                                         |
| <b>bajar lápiz</b>                     | El objeto que ejecuta esta instrucción comienza a dejar un trazo con su movimiento.                                         |
| <b>subir lápiz</b>                     | El objeto que ejecuta esta instrucción deja de dejar un trazo con su movimiento.                                            |
| <b>fijar color de lápiz a</b>          | Se indica el color del trazo que dejará el lápiz.                                                                           |
| <b>cambiar color del lápiz por</b>     | Se modifica incrementando o decrementando el color del trazo que dejará el lápiz. Admite 200 números diferentes de colores. |
| <b>fijar color de lápiz a</b>          | Se indica el color del trazo que dejará el lápiz. Admite 200 números diferentes de colores.                                 |
| <b>cambiar intensidad de lápiz por</b> | Se modifica incrementando o decrementando la intensidad del color del trazo que dejará el lápiz. Admite valores de 0 a 100. |
| <b>fijar intensidad de lápiz a</b>     | Se indica la intensidad del color del trazo que dejará el lápiz. Admite valores de 0 a 100.                                 |
| <b>cambiar tamaño de lápiz por</b>     | Se modifica incrementando o decrementando el grosor del trazo del lápiz. Sólo admite valores positivos.                     |
| <b>fijar tamaño de lápiz a</b>         | Se indica el grosor del trazo del lápiz. Sólo admite valores positivos.                                                     |
| <b>señalar</b>                         | El objeto deja una huella en su posición actual.                                                                            |

Estos bloques nos van a permitir dibujar figuras, cambiar color, intensidad, dibujar recorridos, etc.

### 10.4.1 El bloque borrar



El bloque "borrar" sirve para eliminar los trazos dejados por todos los objetos en el escenario.

Es importante iniciar los programas que utilizan lápiz con este bloque, de esta forma no se acumularán trazos entre una ejecución y otra del programa.

### 10.4.2 El bloque bajar lápiz



Cuando introducimos este bloque en un programa, el objeto se convierte en un lápiz que dibuja por el espacio de trabajo a medida que se va moviendo.

Puedes probar con el siguiente algoritmo



### 10.4.3 El bloque subir lápiz

Este bloque hace que el objeto deje de dibujar.

Puedes comprobar el funcionamiento con el siguiente algoritmo



### 10.4.4 El bloque fijar color de lápiz a...

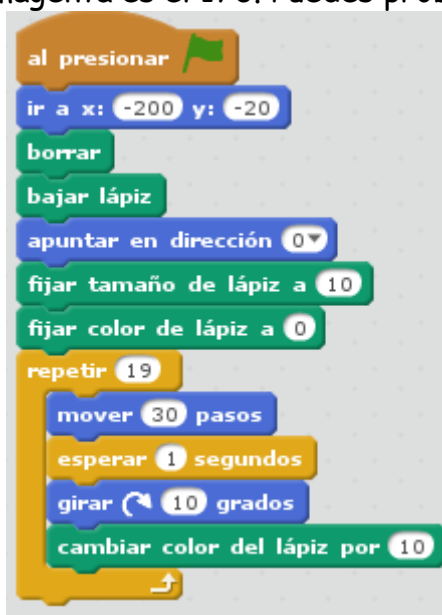


Este bloque sirve para cambiar el color del lápiz

### 10.5.5 Los bloques cambiar y fijar color de lápiz



Estos bloques asignan un color en función del valor numérico asignado. Hay 200 valores posibles de colores, por ejemplo: el rojo es el 0, el verde es el 70, el azul es el 130 o el magenta es el 170. Puedes probar como funciona con el siguiente algoritmo.



#### 10.4.6 Los bloques cambiar y fijar intensidad de lápiz

cambiar intensidad de lápiz por 10

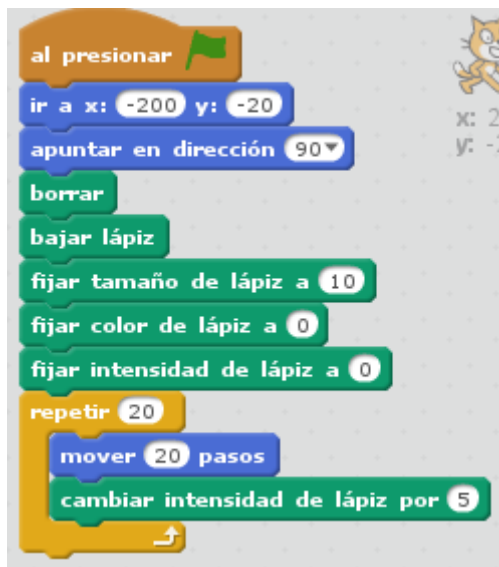
Estos bloques cambian la intensidad del color del trazo.

fijar intensidad de lápiz a 50

El bloque "fijar intensidad" asigna la intensidad indicada y el bloque "cambiar intensidad" incrementa o decrementa el valor de la intensidad en la cantidad indicada.

El valor de la intensidad por defecto es 50, scratch permite valores entre 0 y 100. Los valores cercanos al cero aproximan el color al negro y los cercanos a 100 al blanco.

Puedes comprobar las distintas intensidades con el siguiente algoritmo



#### 10.4.6 Los bloques cambiar y fijar el tamaño de lápiz

El bloque fijar fija un grosor de lápiz, mientras que el bloque cambiar hace que se incremente o reduzca en la cantidad que se indique, teniendo en cuenta que el valor mínimo es 1 y, aunque no hay máximo, un grosor de 480 ocuparía toda la pantalla.



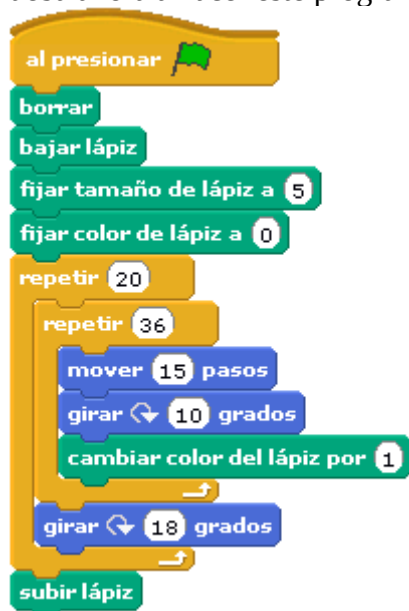
#### 10.4.7 El bloque sellar

El bloque sellar imprime una huella o estela del objeto por donde se mueve el objeto



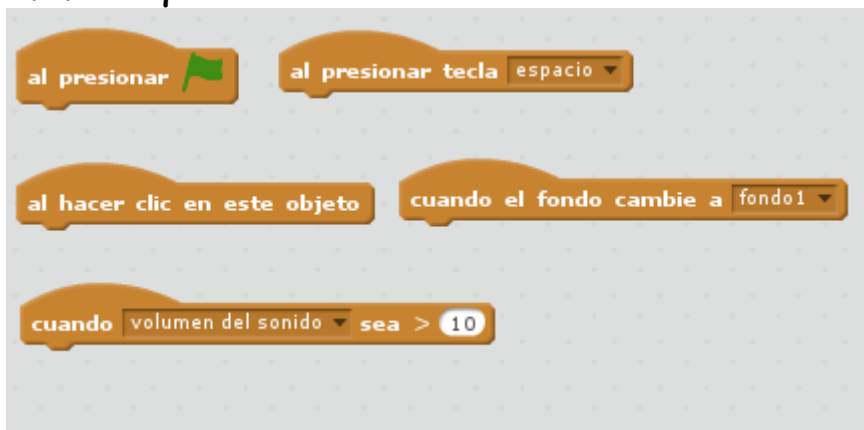
#### 10.4.8 Ejemplo de programa con herramientas de lápiz

Prueba ahora a hacer este programa con el objeto “pencil” y plantea variaciones sobre el



## 10.5 Los bloques de eventos

### 10.5.1 Bloques de inicio



Los bloques de inicio tienen una forma especial, no pueden suceder a ningún otro bloque y solo sirven para empezar una acción en un objeto o fondo. El desencadenante del inicio de la acción o evento puede ser:

Presionar la bandera de inicio de scratch

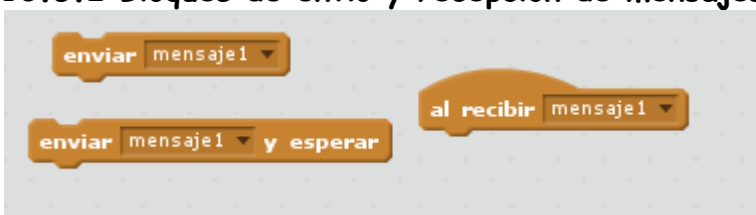
Presionar una tecla en el teclado

Clicar con el ratón un objeto

Un cambio en el fondo

Un cambio en el sonido. tiempo....

### 10.5.2 Bloques de envío y recepción de mensajes



Los bloques de envío y recepción de mensajes se emplean en programas complejos y en programas con varios objetos, sirven para estructurar el programa, en lugar de hacer un algoritmo muy largo hacemos varios algoritmos y algunos de ellos comienzan al recibirse ciertos mensajes, el envío se corresponde con cierto evento.

Puedes entenderlo mejor con este ejemplo:



Escoge el fondo "blue sky" y el objeto "bat 2"

Luego en el programa del objeto bat crea un mensaje que se llame "izquierda", otro que se llame derecha y luego programa el siguiente algoritmo

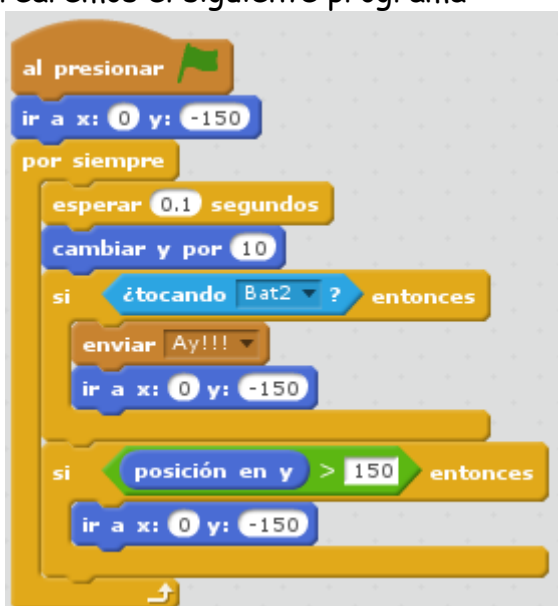


Observa lo que ocurre si en lugar de poner el bloque "enviar mensaje y esperar" pones solo el bloque "enviar mensaje".

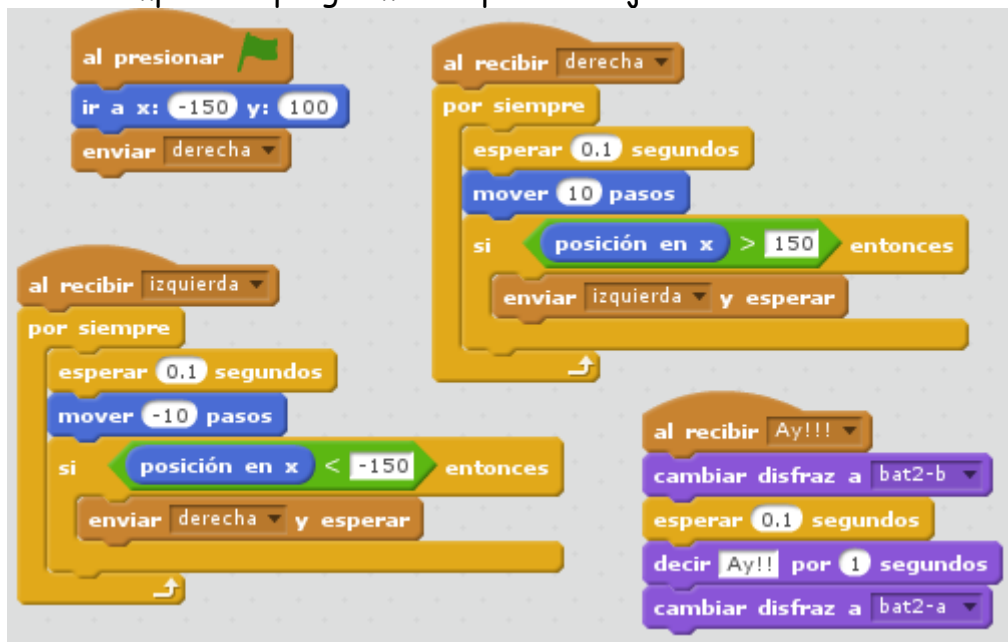
Estos bloques también se utilizan cuando en el programa hay varios objetos y sirven para la interacción y la coordinación entre ellos. La recepción del mensaje es un evento, es decir el inicio de un programa para un objeto determinado

Vamos a añadir un nuevo objeto un nuevo mensaje en el programa anterior.

Tendrás que ir al menú de objetos y buscar el objeto "ball", para este objeto crearemos el siguiente programa



Ahora completa la programación para el objeto bat2



## 10.6 Los bloques de control

Los bloques de control son probablemente los bloques más importantes de scratch y ljunto con los operadores, los más importantes para comprender como funcionan los lenguajes de programación. A lo largo de este curso los hemos utilizado en numerosos algoritmos y ya has podido observar como funcionan y para que sirven.

Ahora vamos a detenernos en cada uno de ellos

### 10.6.1 Los bloques repetir y por siempre



El bloque repetir crea un **bucle** que se repite un número determinado de veces y el bloque por siempre crea un bucle infinito. Como ves en ambos casos estos bloques sirven para englobar otros bloques.

El bloque repetir lo vimos por ejemplo en el apartado 10.1.1 y el bloque por siempre lo vimos en el apartado 10.1.2

### 10.6.2 El bloque esperar ....segundos

Este bloque permite que el programa se detenga durante un cierto tiempo hasta la siguiente acción. Este bloque lo vimos en el apartado 10.2.2

### 10.6.3 Los bloques de condiciones

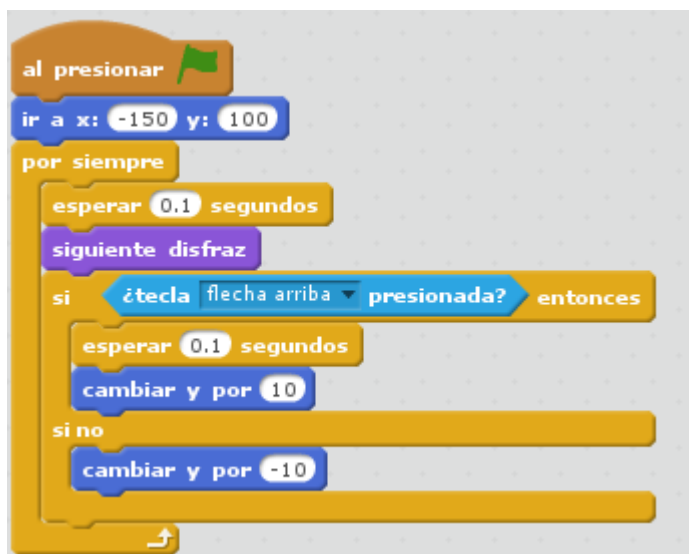


Los bloques condicionales tienen un hueco hexagonal que combina donde se pueden introducir bloques de sensores y operadores

Dentro de ellos los dos primeros, los bloques "si... entonces" y "si ....entonces, si no " son los que llamamos comparadores, hacen que el programa cambie cuando se produzca un determinado suceso.

Utilizamos "si... entonces" cuando queremos que el programa haga algo cuando se cumpla una condición y "si ....entonces, si no " cuando se incumpla.

Prueba a construir el siguiente algoritmo, por ejemplo con el objeto "bat2" y el fondo "blue sky"



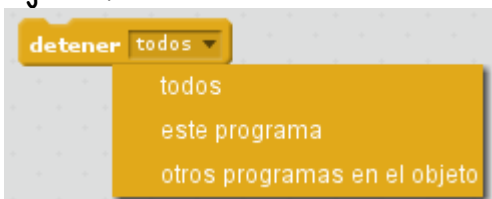
Los bloques "repetir hasta que" y "esperar hasta que " funcionan de una forma antagónica, con el primero se repetirá una acción hasta que se cumpla una condición y con el segundo no se producirá la acción hasta que se cumpla la condición.

Prueba a ver la diferencia entre los dos bloques con los siguientes algoritmos con el objeto "bat2" y el fondo "blue sky"

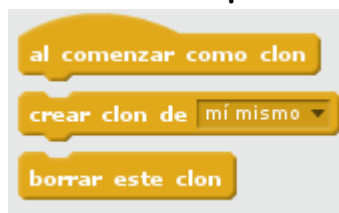


#### 10.6.4 Los bloques detener

Los bloques detener sirven para poner un fin al programa. Puede ser para detener uno o varios programas de un objeto o para detener todos los programas de todos los objetos.



#### 10.6.5 Los bloques clon



Los "clones" son copias de un objeto que nos pueden servir para repetir objetos idénticos en el mismo programa.

Para ver como funciona vamos a utilizar el objeto "ball" y el fondo "blue sky"



## 10.7 Bloques de sensores

Los bloques de sensores tienen gran importancia en las interacciones entre los objetos y en las interacciones del usuario que maneja el programa con el propio programa. Estos bloques son también muy importantes en la programación de robots, ya que estos basan parte de su funcionamiento en el uso de sensores.

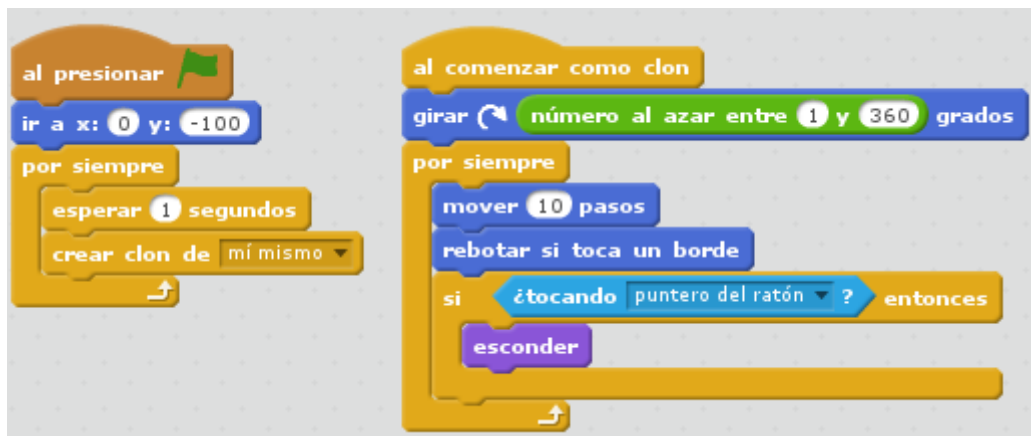
Como puedes comprobar la mayoría de los bloques de sensores tienen la forma hexagonal o redondeada propia de los bloques que se introducen dentro de otro bloque.

### 10.7.1 El sensor "tocando"



Este sensor introduce un evento o cambio en el programa cuando un objeto toca otro objeto, un borde o el puntero del ratón.

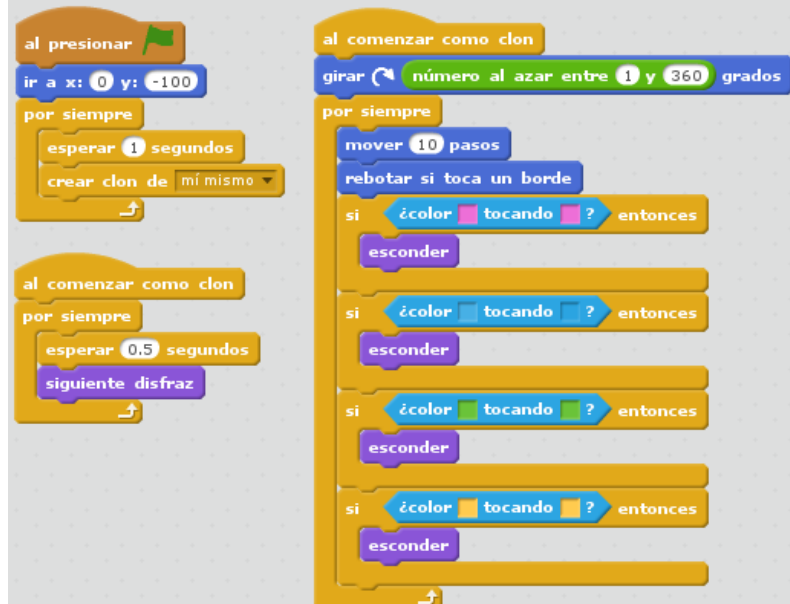
Puedes probar como funciona introduciendo el siguiente programa para el objeto "ball".



### 10.7.2 Los sensores tocando color y color tocando color

Los sensores "tocando color" y "color tocando color" introducen un evento o cambio cuando un objeto toca un color o un objeto de un color determinado toca otro objeto de un color determinado

Puedes crear el siguiente programa para el objeto ball y ver lo que ocurre



### 10.7.3 El sensor "distancia a puntero de ratón"

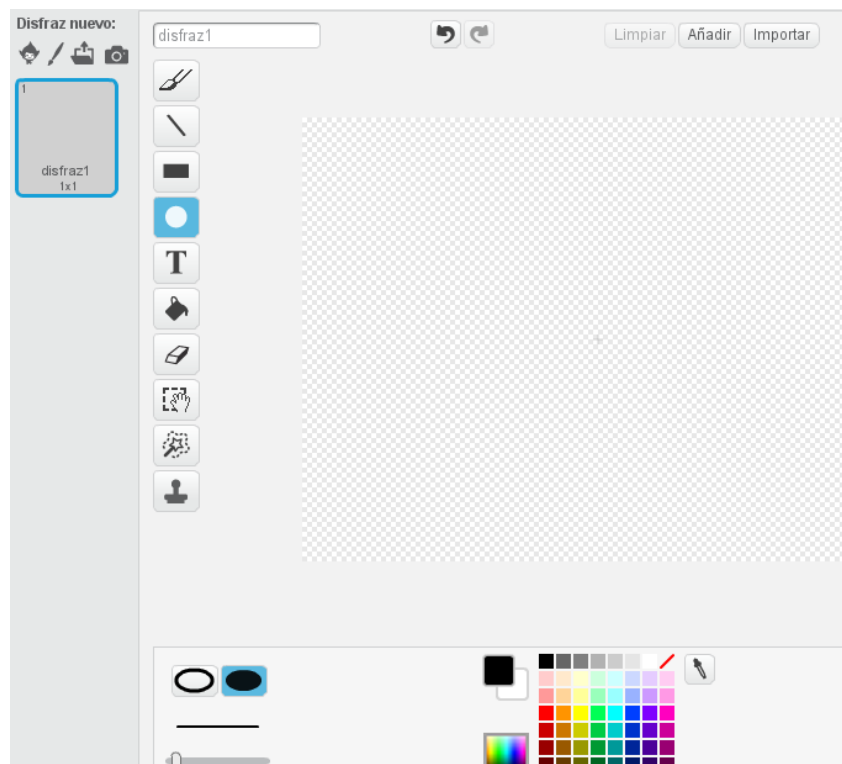
Este sensor introduce un cambio en el programa cuando un objeto se aproxima al puntero del ratón.

Puedes probar el funcionamiento con el siguiente algoritmo



### 10.7.4 El sensor posición x de

Este sensor nos puede servir para que dos objetos se muevan de forma sincronizada, para entenderlo mejor vamos a crear un nuevo objeto con el editor de dibujo al que le vamos a llamar sombra, para ello vamos a utilizar la herramienta del editor de dibujo elipse.



Ahora creamos el siguiente algoritmo para el objeto sombra y observamos lo que pasa



## 10.8 Operadores

El conjunto de bloques de operadores proporciona a scratch infinidad de posibilidades matemáticas.

Este submenú incluye operaciones aritméticas básicas, comparadores, números aleatorios, funciones lógicas.... incluso operaciones con caracteres no numéricos.

### 10.8.1 Los bloques suma, resta, multiplicación y división



Estos bloques nos ofrecen el resultado de la operación



### 10.8.2 Los bloques mayor que, menor que, igual



Los comparadores devuelven el valor verdadero si se cumple la comparación y falso si no se cumple. Ya comprobamos el funcionamiento de este bloque en 10.1.7

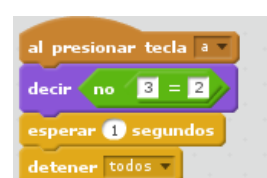
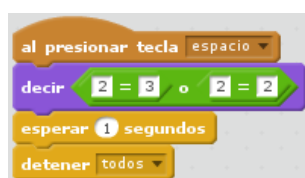
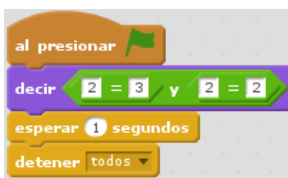
### 10.8.3 Los bloques "y", "o" y "no"

El bloque y devuelve verdadero si las dos condiciones son verdaderas,

El bloque o devuelve verdadero si una de las condiciones es verdadera

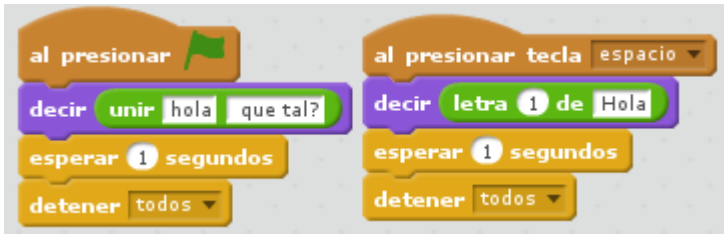
El bloque no devuelve verdadero si la condición es falsa y viceversa

Prueba con los siguientes algoritmos para comprenderlo mejor



#### 10.8.4 Los bloques encadenar y desencadenar

El operador encadenar nos devuelve una cadena de caracteres que es el resultado de unir los distintos elementos de la cadena. El bloque desencadenar nos ofrece un caracter que ocupa una determinada posición de la cadena.



#### 10.8.5 El bloque longitud

El bloque longitud nos devuelve el número de caracteres que contiene una cadena



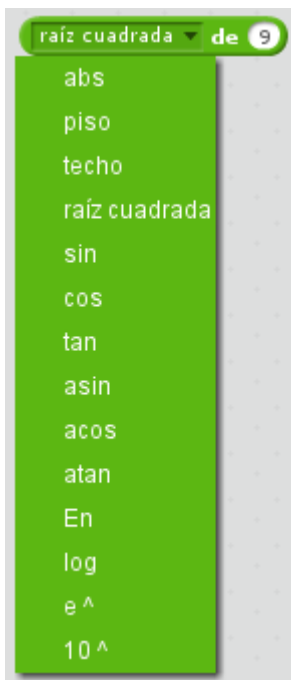
#### 10.8.6 Los bloques mod y redondear



El bloque módulo nos ofrece el resto de una división entre el primer término y el segundo

El bloque redondear nos devuelve el valor redondeado de una operación

#### 10.8.7 El bloque función



Este bloque nos permite seleccionar entre diversas operaciones matemáticas,

## 11 Variables



En el menu de programas, en el submenu datos tenemos la opción de crear variables. Una variable puede almacenar números o cadenas de caracteres. Como su nombre indica el valor de una variable puede variar en el desarrollo del programa.

Podemos usar variables por ejemplo para:

- Almacenar el número de preguntas acertadas en un juego de preguntas.
- Almacenar el total de puntos que tenemos en un juego de cartas.
- Almacenar el número de repeticiones que llevamos en una iteración.
- .....

| OPERADORES                                 |                                                                                      |
|--------------------------------------------|--------------------------------------------------------------------------------------|
| Nueva variable                             | Crear una nueva variable                                                             |
| Borrar una variable                        | Borrar una variable existente                                                        |
| <input checked="" type="checkbox"/> puntos | Variable                                                                             |
| fijar puntos a 0                           | Asigna a la variable el valor indicado                                               |
| cambiar puntos por 1                       | Modifica incrementando o decrementando el valor de la variable con el valor indicado |
| mostrar variable puntos                    | Muestra la variable en el escenario                                                  |
| esconder variable puntos                   | Oculto la variable en el escenario                                                   |

### 11.1.1 Crear una nueva variable

Variable nueva

Nombre de la variable:

☒ Para todos los objetos ☐ Sólo para éste objeto

OK Cancelar

Ojo, lógicamente no podemos hacer 2 variables con el mismo nombre, el programa no lo permite.

Una vez que creamos la variable, el programa genera una serie de bloques relacionados con esa variable.

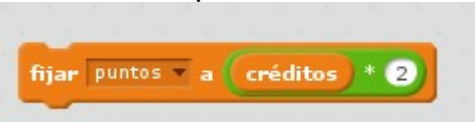


Si hacemos click en la marca que aparece junto a la variable, esta se mostrará en la pantalla en todo momento.

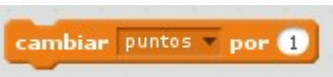
### 11.1.2 El bloque fijar



El bloque fijar le da el valor asignado a la variable, puede ser un valor determinado, pero también puede ser el resultado de una operación matemática



### 11.2.2 El bloque cambiar



Normalmente queremos que el valor de la variable, varíe, para eso hacemos que se modifique su valor ante determinado suceso.

### 11.2.3 Videojuego de rebote con variable

En primer lugar vamos a introducir el objeto paddle en el programa, eliminamos el gato que viene por defecto.

Para que este objeto se mueva horizontalmente por la parte inferior de la pantalla introducimos el programa



Y comprobamos que funcione

En segundo lugar vamos a introducir el objeto "ball" en el programa. Mejora el juego si hacemos el objeto ball algo más pequeño.



(Con esta herramienta se hace el objeto más pequeño)

Creamos una variable que se llame puntos e introducimos el siguiente algoritmo.



Comprobamos el funcionamiento y le introducimos las mejoras que consideremos oportunas.

## 12.1- Listas

En el menu de programas, en el submenu datos tenemos la opción de crear listas.

Una lista es una estructura de datos referenciada por un nombre que permite almacenar un conjunto de datos durante la ejecución del programa.

Los datos pueden ser numéricos o no.

Los datos de una lista reciben el nombre de elementos.

Cada lista debe tener un nombre.

La diferencia entre una variable y una lista es que la variable sólo almacena un valor mientras que la lista puede almacenar varios valores.

La lista nos puede servir para:

- Almacenar las respuestas preguntas y en un juego de preguntas.
- Almacenar el total de puntos que hemos conseguido en cada uno de los niveles de un juego.
- Almacenar una lista de palabras válidas.

| OPERADORES |                                                                        |
|------------|------------------------------------------------------------------------|
|            | Añade una nueva lista al proyecto.                                     |
|            | Borra una lista existente.                                             |
|            | Bloque de la lista.                                                    |
|            | Añade el elemento indicado a la lista.                                 |
|            | Borra el elemento indicado por la posición de la lista.                |
|            | Inserta en la lista el elemento indicado en la posición proporcionada. |
|            | Reemplaza el elemento indicado por la posición en la lista.            |
|            | Devuelve el elemento de la lista que ocupa la posición indicada.       |
|            | Devuelve el número de elementos que contiene la lista.                 |
|            | Operador lógico que devuelve si un elemento está o no en la lista.     |

### 12.1 Utilizando las listas de Scratch para un juego de preguntas

Vamos a crear un juego de preguntas y respuestas utilizando listas

Primero creamos la lista "preguntas".

Lista nueva

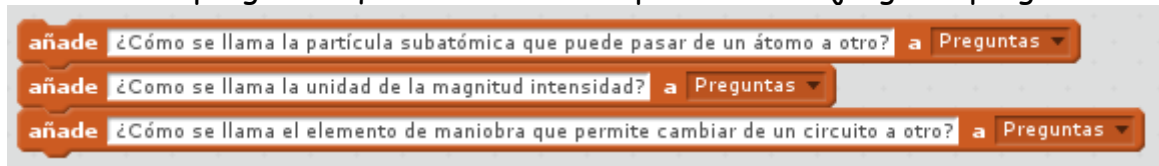
Nombre de la lista:

☒ Para todos los objetos ☐ Sólo para éste objeto

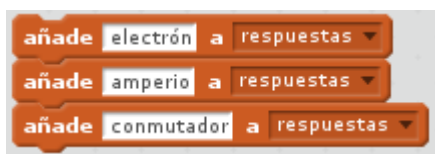
Posteriormente empezamos a crear las preguntas con la variable:



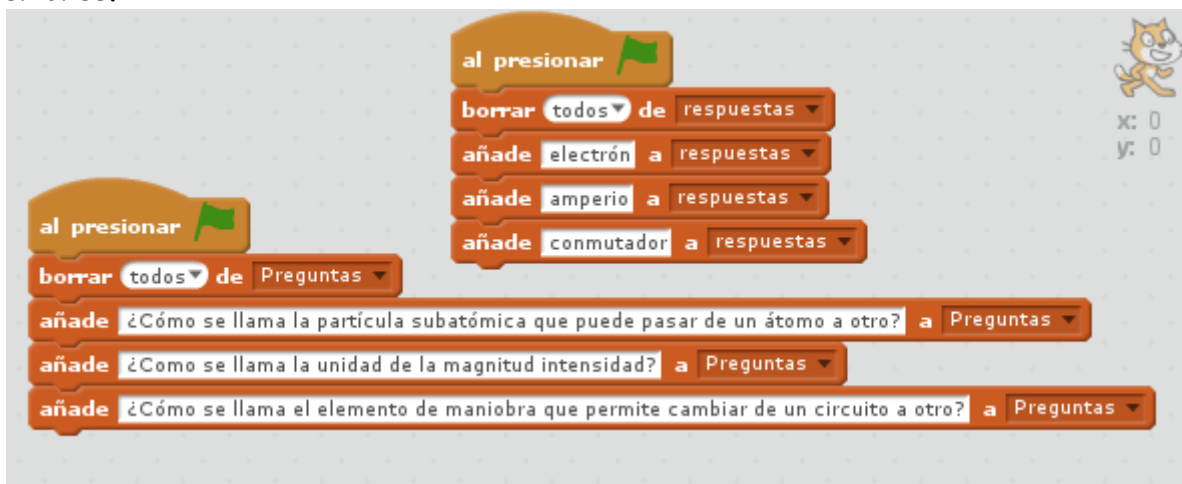
Y escribimos las preguntas que se nos ocurran para nuestro juego de preguntas



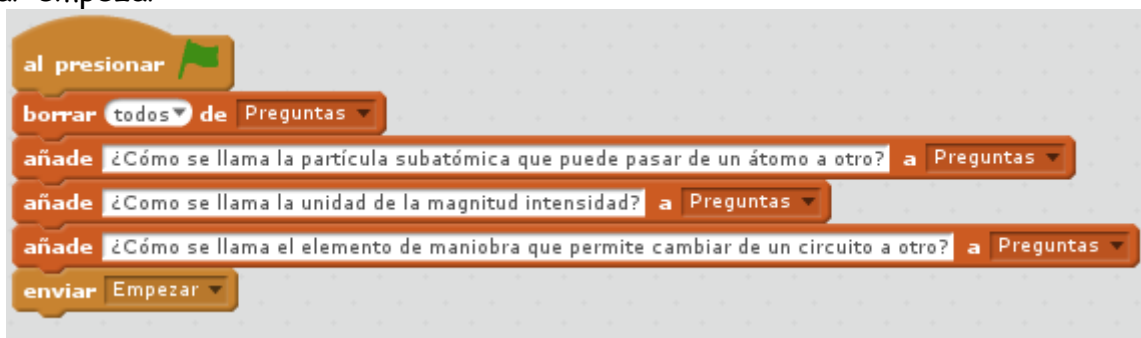
Posteriormente creamos una nueva lista llamada "respuestas" y creamos las respuestas correctas



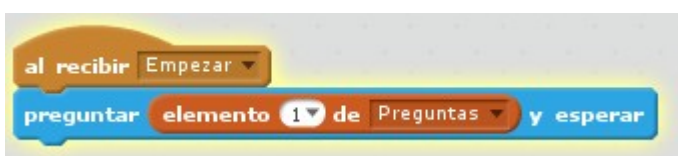
Para que el programa funcione bien debemos de empezar eliminando las listas anteriores.



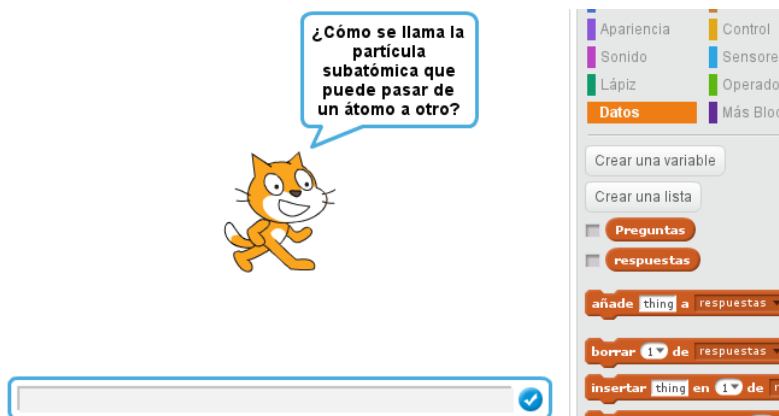
Para que empiece realmente el juego podemos probar a enviar un mensaje, le podemos llamar empezar



Y al recibir el mensaje empezar, el objeto formula la primera pregunta



Vemos que en la pantalla, además de la pregunta, aparece un espacio donde podemos escribir la respuesta...



Pero todavía no hemos hecho la interacción necesaria entre las preguntas y las respuestas. La podemos empezar a hacer de la siguiente manera:



Si ya lo has hecho y funciona intenta tu mism@ acabar el ejercicio o hacer uno a tu manera.