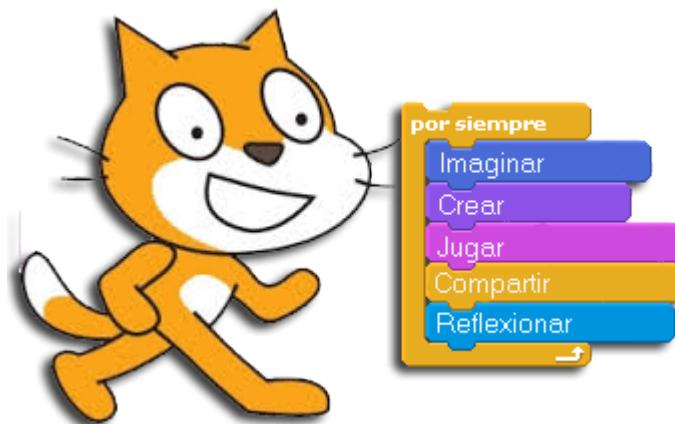


# CURSO SCRATCH



NOMBRE: \_\_\_\_\_

Proyecto:  
“Desarrollo de Habilidades  
Tecnológicas usando Scratch  
y Lego Mindstorm”

# Contenido

Unidad 1.....	- 1 -
¿Qué es Programar? .....	- 1 -
¿Qué es Scratch? .....	- 1 -
Entorno de Trabajo.....	- 2 -
ÁREA DE ETIQUETAS .....	- 2 -
Área de programación .....	- 3 -
Ej. Mi primer Programa en Scratch.....	- 4 -
ESCENARIO .....	- 4 -
Aérea de Sprite.....	- 5 -
¿Como guardar mis proyectos?.....	- 5 -
¿Cómo abrir mis proyectos? .....	- 6 -
TALLER UNIDAD 1.....	- 7 -
Unidad 2.....	- 9 -
¿Como ordenar mis datos?.....	- 9 -
¿Qué es una Variable? .....	- 9 -
Características de las Variables .....	- 10 -
Creación de una variable .....	- 10 -
TALLER UNIDAD 2.....	- 12 -
Unidad 3.....	- 13 -
¿Qué son algoritmos? .....	- 13 -
Algoritmos en Computadoras .....	- 13 -
Algoritmos en Scratch .....	- 13 -
Algoritmos con decisiones .....	- 14 -
Condicionales .....	- 14 -
Repeticiones .....	- 15 -
Algoritmos con decisiones .....	- 15 -
Etiquetas de control.....	- 16 -
Descripción de etiquetas .....	- 16 -
Algoritmos en Scratch .....	- 17 -
Ejemplo: Sumar click.....	- 17 -
TALLER UNIDAD 3.....	- 18 -
Unidad 4.....	- 20 -
Listas.....	- 20 -
Cómo creamos una lista es scratch: .....	- 20 -
Scratch ofrece los siguientes bloques de control para las listas.....	- 21 -
Ejemplo con listas .....	- 21 -

TALLER UNIDAD 4.....	- 24 -
Unidad 5.....	- 26 -
Eventos.....	- 26 -
Eventos de teclado.....	- 26 -
Controles .....	- 27 -
Sensores .....	- 27 -
Animaciones .....	- 28 -
Movimiento .....	- 28 -
Sonido .....	- 28 -
TALLER UNIDAD 5.....	- 29 -
Unidad 6.....	- 31 -
Por qué usar lógica .....	- 31 -
EXPRESIÓN BOOLEANA .....	- 31 -
Algunas etiquetas booleanas .....	- 31 -
Hilos .....	- 32 -
Multihilos .....	- 33 -
Eventos.....	- 33 -
Aleatoriedad.....	- 33 -
TALLER UNIDAD 6.....	- 35 -
<b>ACTIVIDADES .....</b>	<b>- 36 -</b>

## Unidad 1



### ¿Qué es Programar?

- Programar es decirle a tu computadora que cosas quieras que haga por ti, es hablar en su idioma y enseñarle o decirle todo lo que necesites que realice.
- Es realizar algo a través de una serie de pasos detallados. Por ejemplo, el acto de dormir sigue un orden lógico:
- Luego escribiremos estos pasos. Podría ser en español, en inglés o cualquier otro idioma, pero las instrucciones seguirán siendo las mismas



### ¿Qué es Scratch?

- Scratch es un lenguaje de programación que facilita crear historias interactivas, juegos y animaciones.
- Puedes compartir tus creaciones con otras personas en la web.
- ¡Te convertirás en un programador!



## Entorno de Trabajo



## ÁREA DE ETIQUETAS



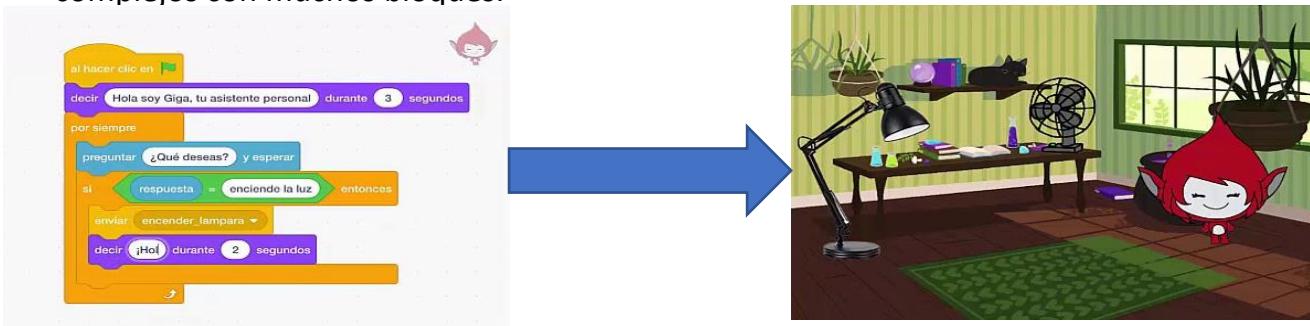
En esta área nos muestra el conjunto de instrucciones que podremos utilizar en la creación de programas. En total contamos con 8 elementos con sus respectivos colores.

- AZUL para el MOVIMIENTO
- Morado Claro para la APARIENCIA
- Lila para SONIDO
- Amarillo para EVENTOS
- Marrón Claro para CONTROL
- Azul Claro para SENsoRES
- Verde para OPERADORES
- Naranja para DATOS(Variables)
- Rosado para MIS BLOQUES



## Área de programación

- Muestra todos los programas asociados al objeto activo.
- Los programas se crean arrastrando los bloques de la parte izquierda a esta área.
- Se puede crear desde programas simples de unos cuantos bloques como programas más complejos con muchos bloques.



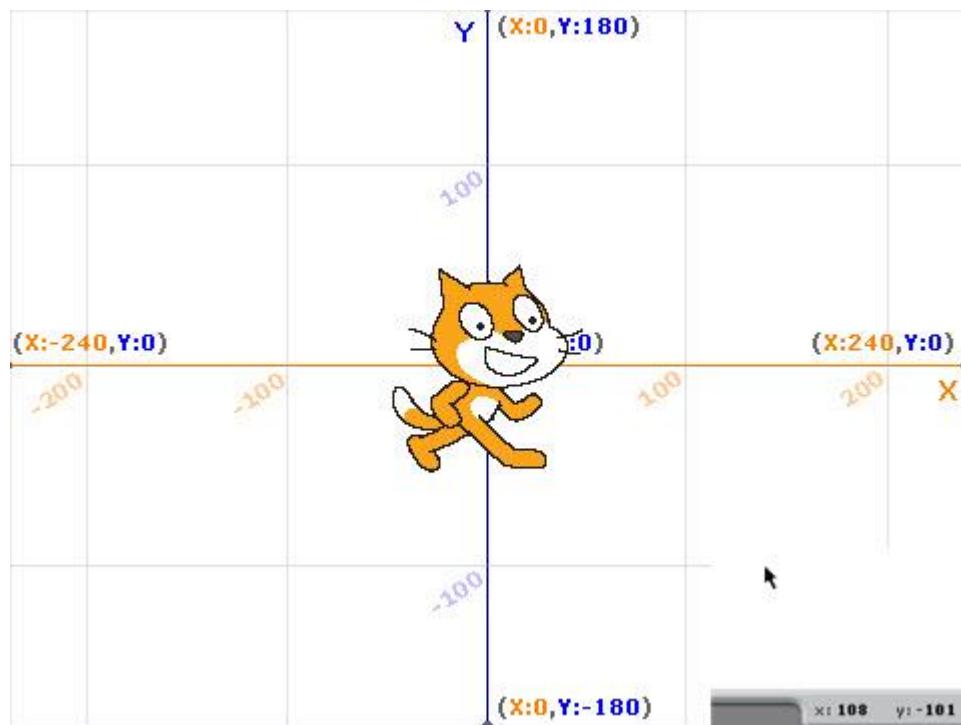
## Ej. Mi primer Programa en Scratch.

- Haremos que el Gatito Scratch salude diciendo: ¡Hola Mundo! Durante 5 segundos.



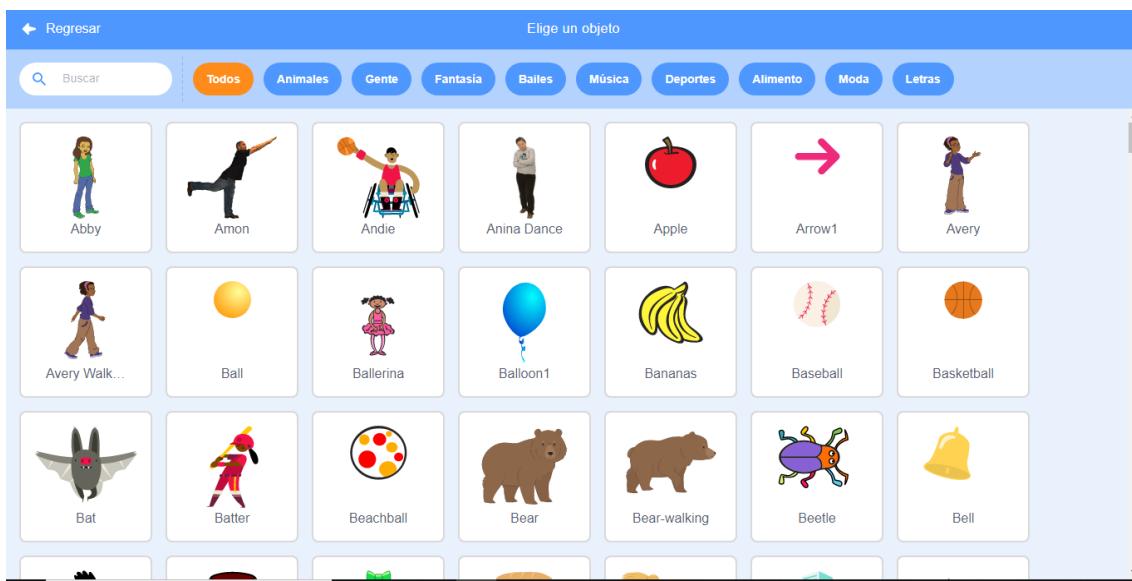
## ESCENARIO

- Es donde podemos ver nuestro proyecto en ejecución.
- El **escenario** de Scratch tiene unas dimensiones de **480 x 360 píxeles**.
- Los objetos se muestran y mueven por el escenario, un área especialmente importante del entorno.
- Podemos considerar el escenario como una superficie (área plana con dos dimensiones o dos ejes).
- Se trata de una superficie cuyas coordenadas en el eje horizontal (X) va desde la posición -240 (izquierda) hasta la posición 240 (derecha). Por su parte, el eje vertical (Y) va desde la posición -180 (abajo) hasta la posición 180 (arriba).
- La posición central del escenario será por tanto el punto (0,0).



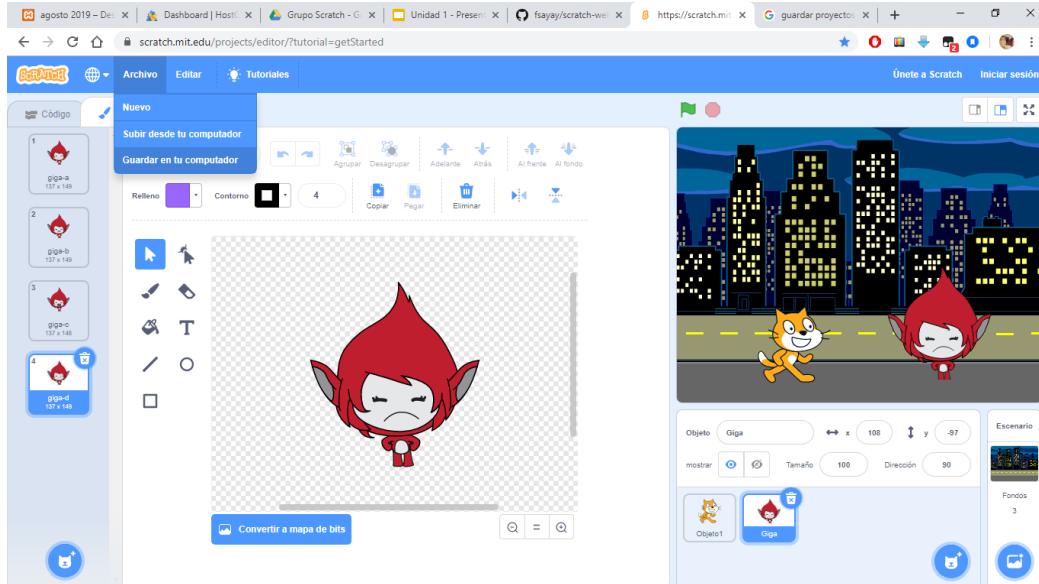
## Aérea de Sprite

- Cada objeto en Scratch también recibe el nombre de Sprite.
- Al que se tiene acceso desde la biblioteca de objetos que tiene Scratch.
- Scratch incorpora un conjunto de imágenes que pueden ser utilizadas como objetos dentro de tu proyecto.

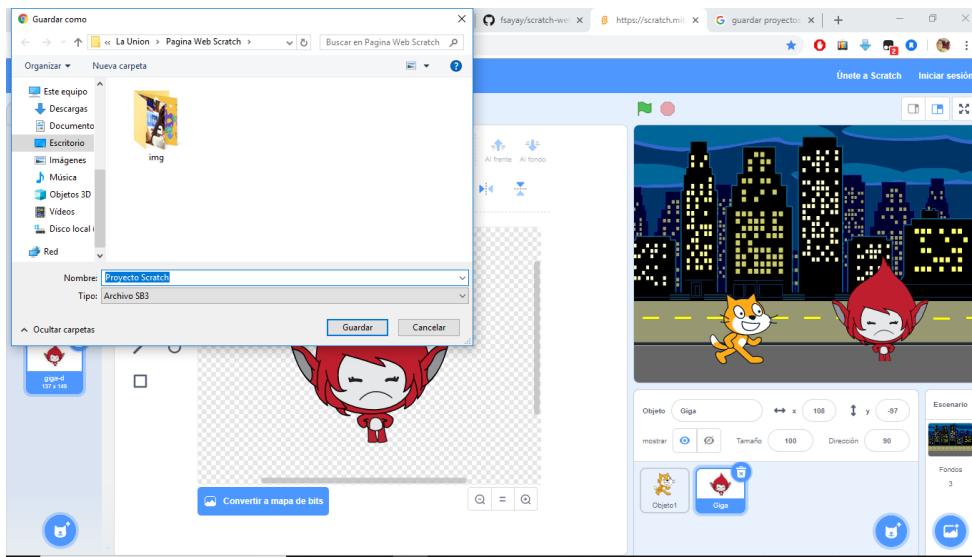


## ¿Cómo guardar mis proyectos?

- Primero hacemos click en ARCHIVO y luego damos clic en GUARDAR EN TU COMPUTADOR.

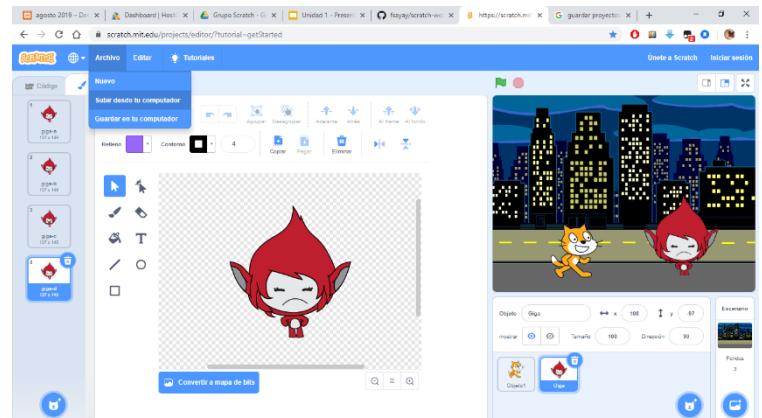


- Por último, elegimos la carpeta donde queremos que se guarde nuestro proyecto y listo.

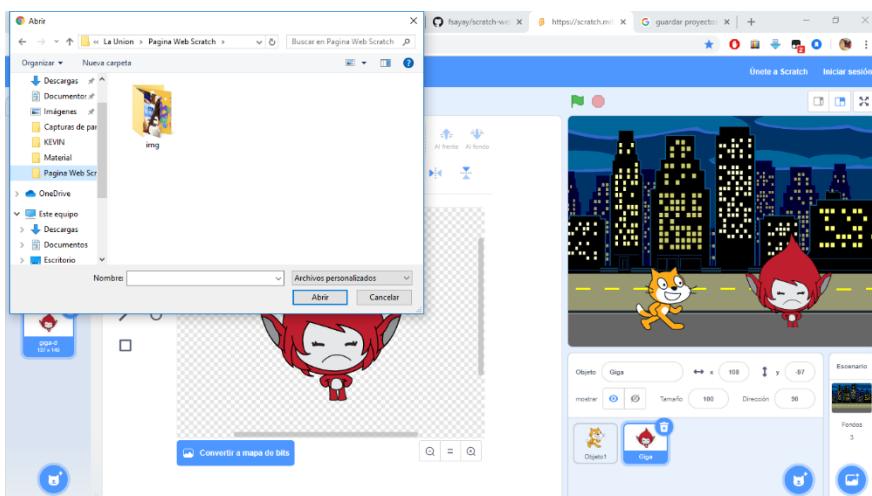


## ¿Cómo abrir mis proyectos?

- Primero hacemos click en ARCHIVO y luego damos click en SUBIR DESDE TU COMPUTADOR.



- Elegimos nuestro proyecto y damos en abrir.

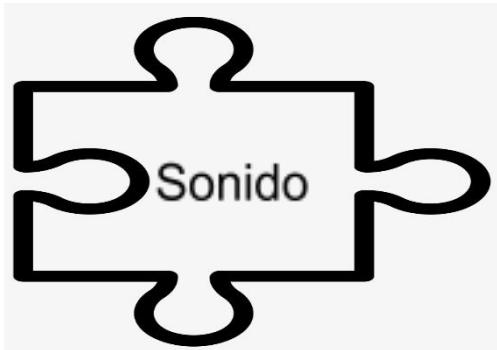
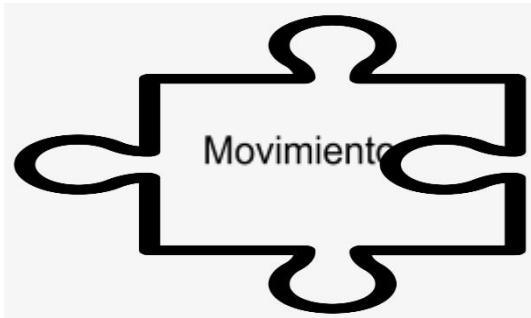


Fecha:  
Nombre:

## TALLER UNIDAD 1 SCRATCH

### 1. Colorea los siguientes recuadros según lo indicado en las etiquetas:

- Azul-Movimiento
- Morado-Apariencia
- Lila-Sonido
- Amarillo-Evento



### 2. Unir con línea la etiqueta con su respectivo color

Marrón Claro

Mis Bloques

Azul Claro

Operadores

Verde

Datos

Naranja

Sensores

Rosado

Control

**3. Encierra las etiquetas que corresponden a Control.**



**4. Listas dos etiquetas de movimiento.**

---



---

**5. Coloca la letra A para las etiquetas de Apariencia.**

Cambiar Efecto

Fondo siguiente

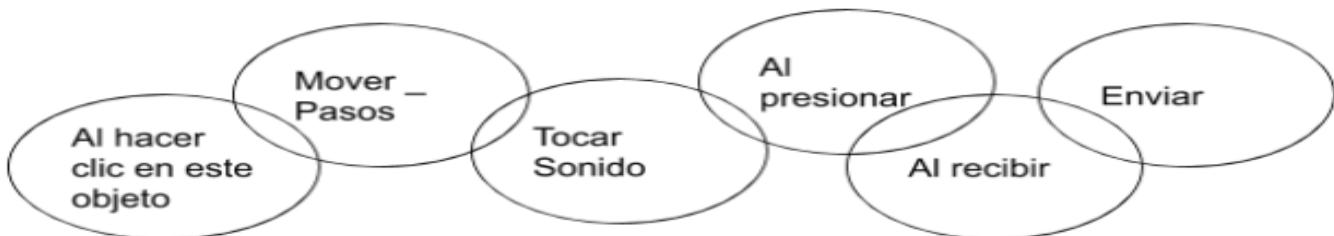
Pensar por segundo

Enviar y esperar

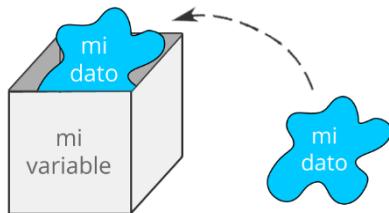
Apuntar en dirección

Pensar

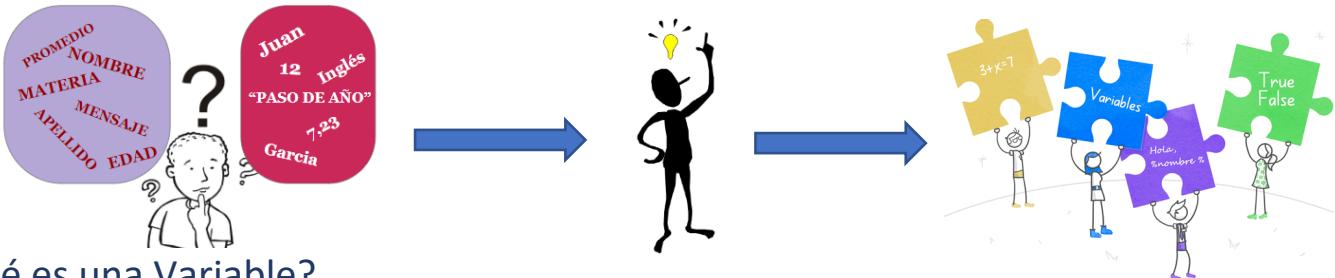
**6. Coloca un asterisco para las etiquetas de Eventos.**



## Unidad 2

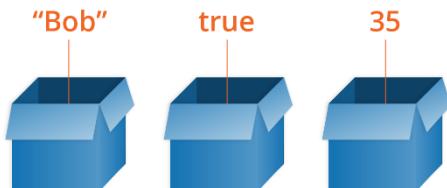


¿Como ordenar mis datos?

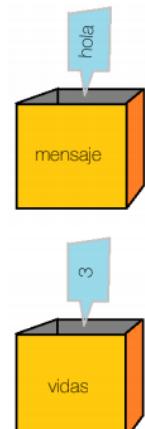
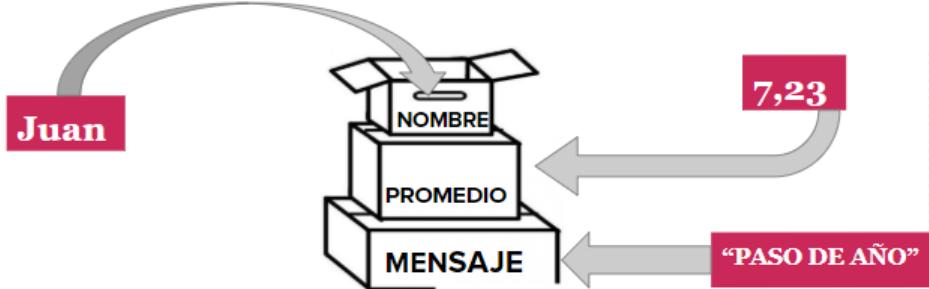


¿Qué es una Variable?

- Una variable es donde se guarda (y se recupera) datos que se utilizan en un programa.
- Las variables solo pueden contener un valor a la vez, Estos valores pueden ser números, cadenas o valores verdaderos y falsos que los conocemos como “booleanos”.



- Puedes cambiar el valor de una variable. Utiliza nombres fáciles que tengan relación con el contenido que almacena la variable para trabajar de manera más sencilla.
- Cuando creamos una nueva variable siempre se nos pregunta si queremos una variable para todos los objetos o una variable para este objeto:



- La variable "Sólo para este objeto" o local sólo es visible y se puede modificar por el objeto que la tiene asignada.
- La variable "Para todos los objetos" o global es visible y se puede modificar por cualquier objeto del proyecto.

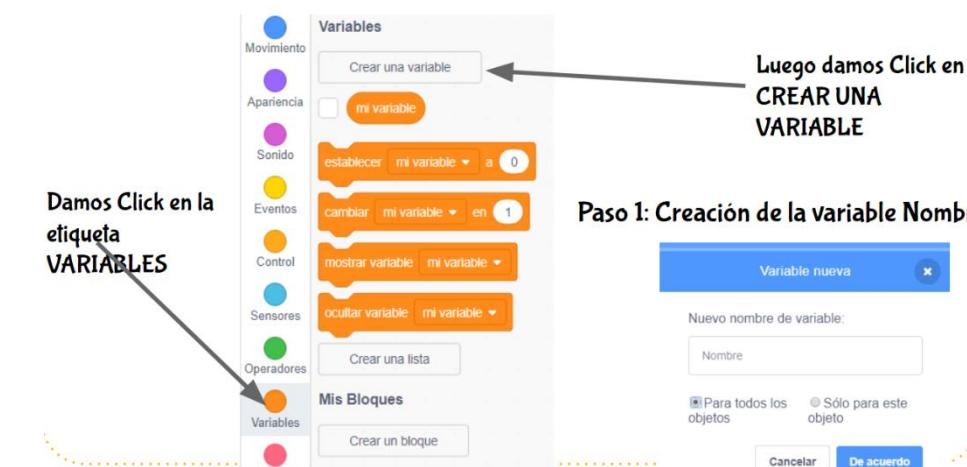
## Características de las Variables

Para acceder al bloque de las variables tenemos nuestro objeto color naranja en nuestra paleta de bloques.



- Crea una nueva VARIABLE.
- Nombre de mi variable.
- Asigna un valor a mi variable.
- Modifica el valor de mi variable.
- Muestra la variable en el escenario.
- Oculta la variable del escenario.

## Creación de una variable



Una vez creado la variable Nombre



Paso 1: Creación de la variable Nombre

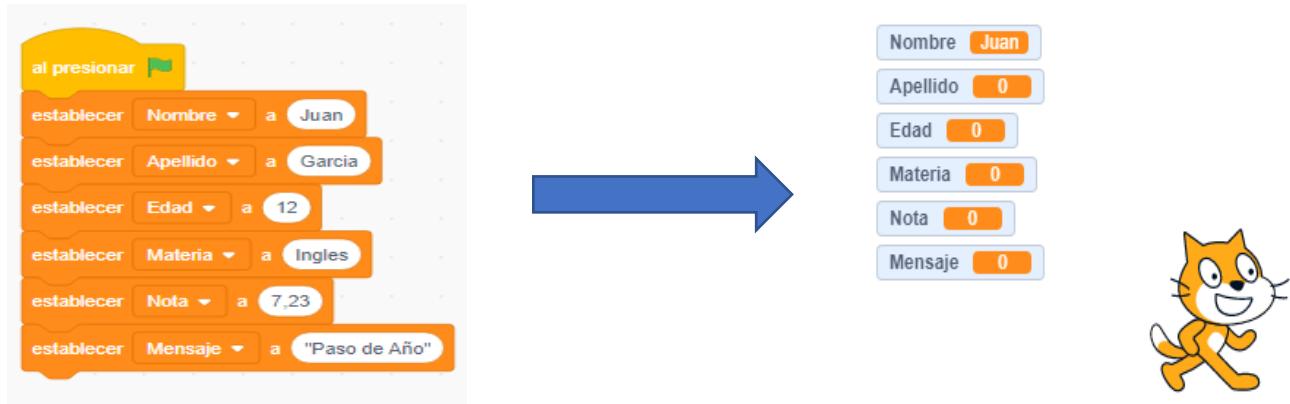


Paso 2: Inicialización de la variable Nombre al inicio del programa.



Repetir los 2 pasos para las otras variables:  
Apellido, Edad, Materia, Nota, Mensaje.

Al final tendremos algo como esto Con su respectiva Inicialización y listo para ser USADAS dentro del programa.

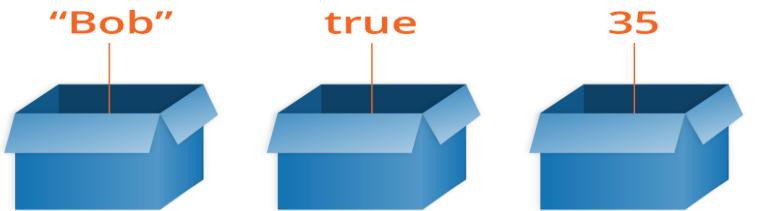


Fecha:  
Nombre:

## TALLER UNIDAD 2

### VARIABLES

#### 1. Identifique el tipo de variable en Scratch:



#### 2. Unir con línea según corresponda.



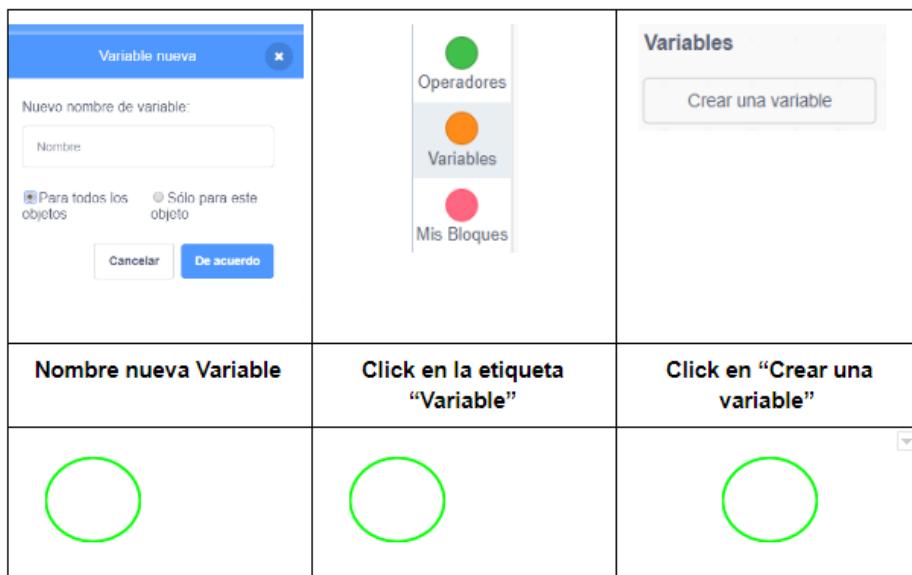
Muestra “mi variable” en el escenario

Oculta “mi variable” del escenario

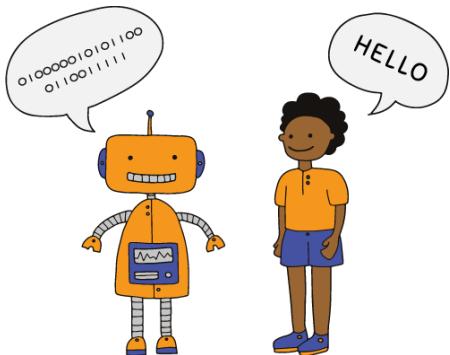
Asigna un valor a “mi variable”

Modifica el valor de “mi variable”

#### 3. Ordene los pasos para crear una nueva variable.



## Unidad 3



### ¿Qué son algoritmos?

Pasos ordenados, finitos, no ambiguos que describen cómo realizar una tarea. Los algoritmos forman programas.



#### Pasos para lavarse las manos

1. Abrir la llave de agua
2. Mojarse las manos
3. Aplicar el jabón
4. Enjugar las manos
5. Secarse las manos

### Algoritmos en Computadoras

Los algoritmos nos ayudan a realizar tareas repetitivas, complicadas o largas. Las computadoras realizan lo que les pidamos y lo harán con gran rapidez.

Por ejemplo:

Una persona se puede demorar en realizar la multiplicación de  $4500 * 8520$ , pero una computadora lo realiza al instante

Las computadoras no entienden nuestro lenguaje, usamos algoritmos para decirles qué deben hacer. Por eso los algoritmos especifican pasos simples, precisos y sin ambigüedades.

Usamos lenguajes específicos que las computadoras entienden, nosotros usaremos Scratch.

### Algoritmos en Scratch

¿En dónde empieza el algoritmo?

Existen varias opciones para empezar, entre ellas:

- Cuando se haga click en la bandera de Scratch

- Cuando presionemos una tecla, por ejemplo, la barra de espacio
- Cuando hagamos click sobre un Sprite.



## Algoritmos con decisiones

Existen instrucciones que nos indican que realizar en caso de una condición

- Si ocurre esto, entonces realizar aquello
- Si ocurre esto, entonces realizar aquello, sino esto.

## Condicionales

Nos permiten realizar diferentes acciones según un valor, verdadero o falso

Ejemplo:

- Si el vaso no tiene agua, entonces llenarlo
- Si tienes hambre, entonces aliméntate, sino continúa estudiando.

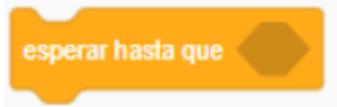


Permiten tomar diferentes caminos según la condición:

Si se cumple la condición dada se ejecuta lo que está dentro de la etiqueta, si no se cumple se ejecutará lo que le sigue a la etiqueta. O se ejecuta lo que se coloque en la sección de sí no.



Los bloques se detienen hasta que se cumpla la condición dada



## Repeticiones

- Repetir una acción hasta que se cumpla una condición.
  - Mientras ocurra esto, sigue repitiendo aquello.

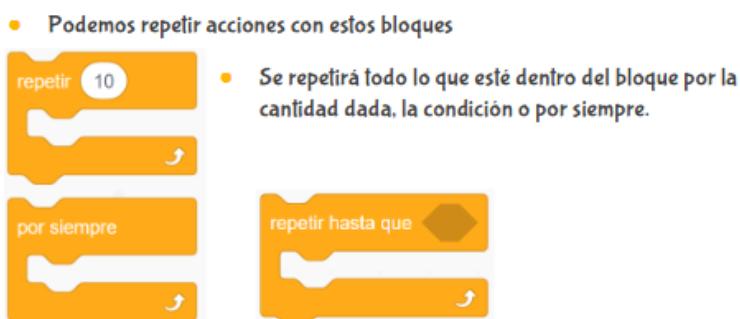
Ejemplos:

Mientras tengas hambre, aliméntate.  
 Mientras tus manos están mojadas, sécate las manos  
 Mientras no termines la tarea, continúa trabajando  
 Mientras el vaso no esté lleno, seguir llenando

- Repetir acción cierta cantidad de veces.
  - Repetir esto 10 veces.
  - Repetir esto 5 veces.
  - Repetir esto n veces

Ejemplos:

Saltar 10 veces  
 Dar 5 pasos  
 Escribir mensaje 20 veces



## Algoritmos con decisiones

Combinando estas herramientas podemos crear algoritmos con mayor funcionalidad.

Tienes hambre?

Si:

    Buscar comida en el refrigerador

    Hay comida?

        Si:

            Preparar comida

        No:

            Salir a comprar

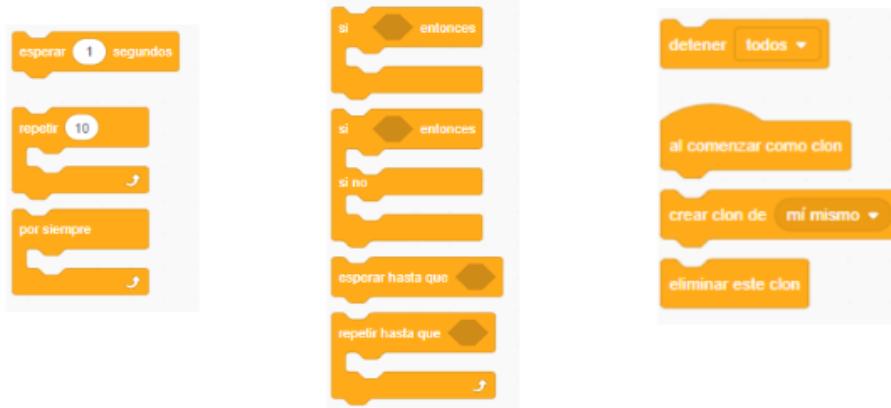
    Comer

    No:

        Satisfecho

## Etiquetas de control

Scratch ofrece las siguientes etiquetas de control:



### Descripción de etiquetas



Se detienen los bloques en ese punto por la cantidad de segundos ingresada.



Repite lo que esté dentro de esta etiqueta ya sea por una cantidad dada o por siempre.

Si se cumple la condición dada se ejecuta lo que está dentro de la etiqueta, si no se cumple se ejecutará lo que le sigue a la etiqueta. O se ejecuta lo que se coloque en la sección de sí no.



Los bloques se detienen hasta que se cumpla la condición dada



Se ejecuta lo que se encuentre dentro de la etiqueta hasta que la condición que se ingrese sea falsa.



Detiene la ejecución de los sorites que se especifiquen, o todos.



Permiten crear copias de un Sprite, decidir qué ocurre cuando se crea una copia y eliminar a esa copia.

## Algoritmos en Scratch

Nos permiten crear animaciones, juegos, programas interactivos. Usualmente requieren la etiqueta por siempre, ya que siempre se espera a algún evento.



Se usa la repetición para ahorrar instrucciones repetitivas. Se gira y se sigue moviendo.

### Ejemplo: Sumar click.

La variable contadora guarda nuestra cuenta. Se espera a que exista un click en el Sprite. Cuando esto ocurra, le sumamos 1 a nuestra variable y la mostramos.



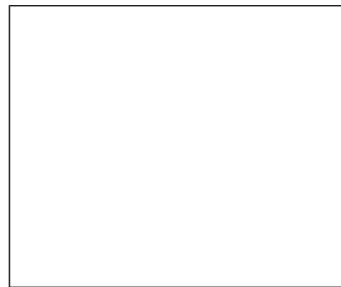
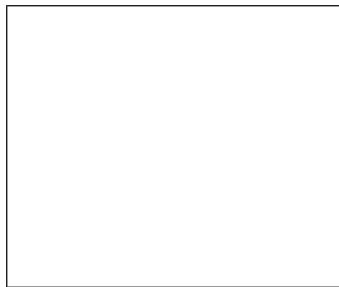
Fecha:

Nombre:

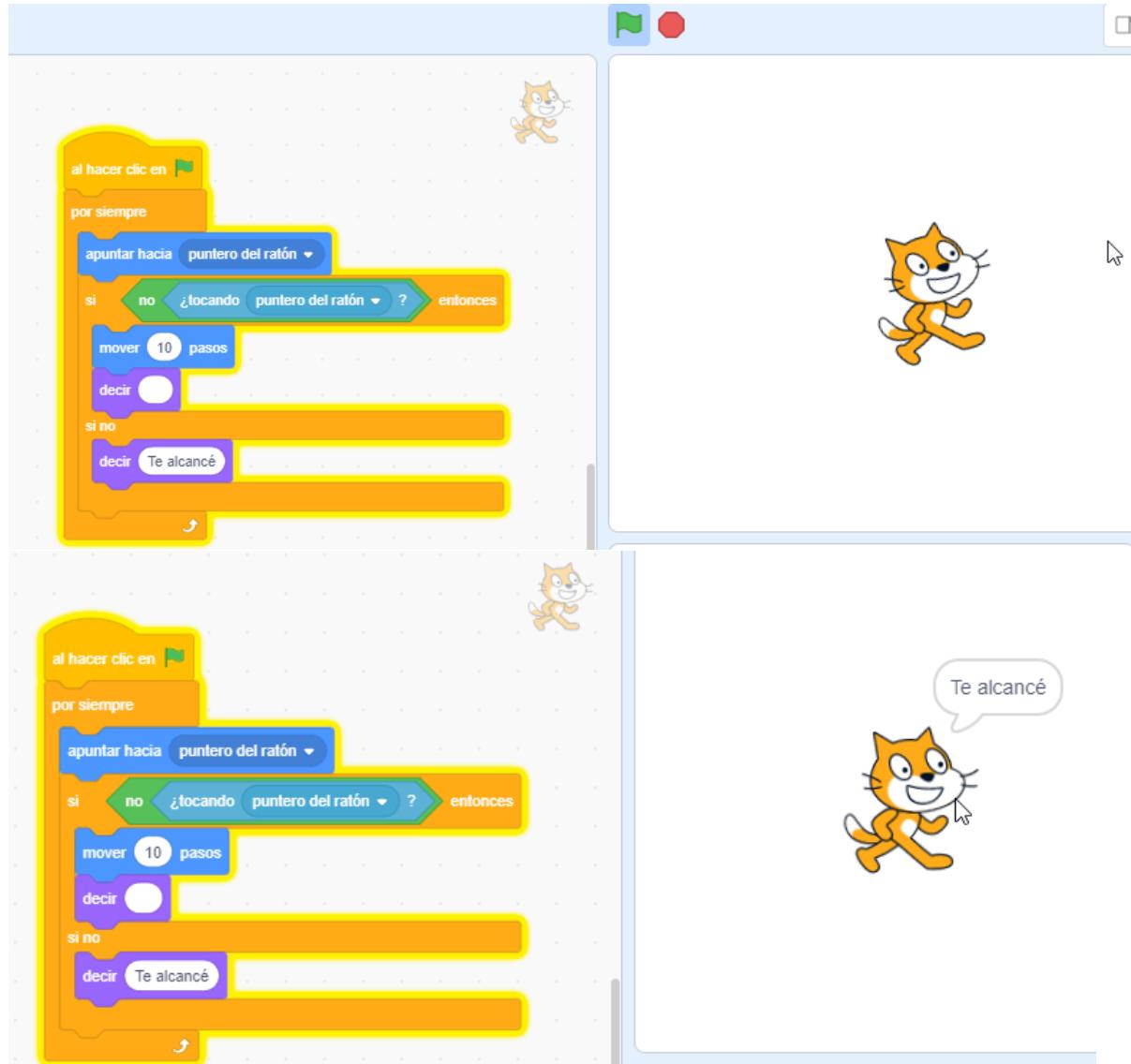
## TALLER UNIDAD 3 ALGORITMOS

- 1. Escriba un algoritmo para ejecutar un programa en Scratch:**

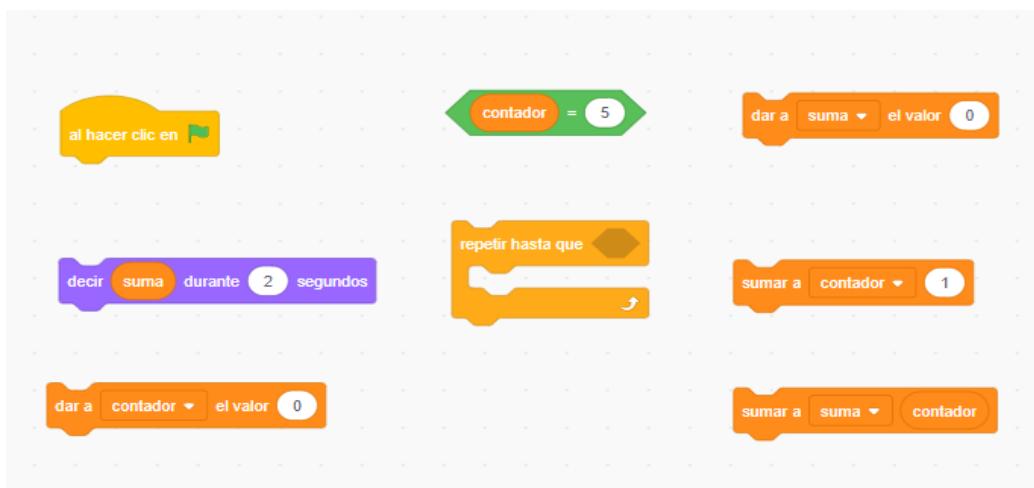
- 2. Dibuje 4 etiquetas de control de Scratch**



3. ¿Qué realiza el siguiente algoritmo? ¿Que realizan las etiquetas?



4. Ordene las etiquetas para crear un algoritmo que muestre la suma del 1 al 5.



## Unidad 4



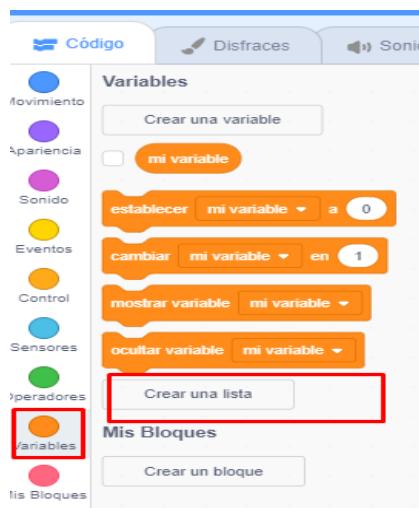
### Listas.

- Es una herramienta que se puede utilizar para almacenar múltiples piezas a la vez.
- Una lista se la reconoce por un nombre que permite almacenar varios valores

Tenemos las siguientes operaciones para manipularlas:

- Insertar
- Añadir
- Borrar
- Mostrar
- Reemplazar

Cómo creamos una lista es scratch:



No se ve el contenido de la palabra debajo del círculo naranja

1. Variables
2. Crear una lista
3. Creamos la lista Meses.
4. Puede ser Global o Local

## Scratch ofrece los siguientes bloques de control para las listas

elemento 1 de Meses ▾

Devuelve el elemento indicado. En este caso nos devuelve el primer elemento de la lista

elemento # cosa en Meses ▾

Busca en la lista por el nombre deseado

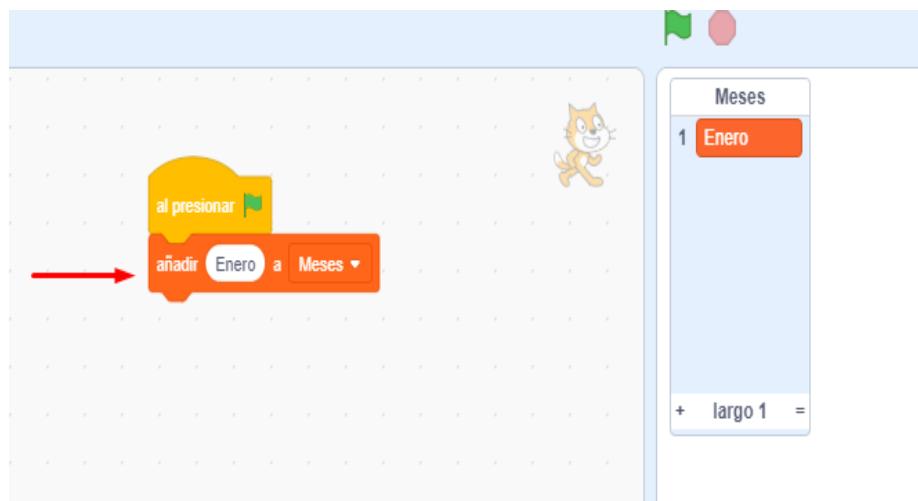
longitud de Meses ▾

Cantidad de elementos que posee la lista

¿ Meses ▾ contiene cosa ?

Pregunta si la lista tiene un determinado valor

## Ejemplo con listas



¡Agreguemos meses a nuestra lista!

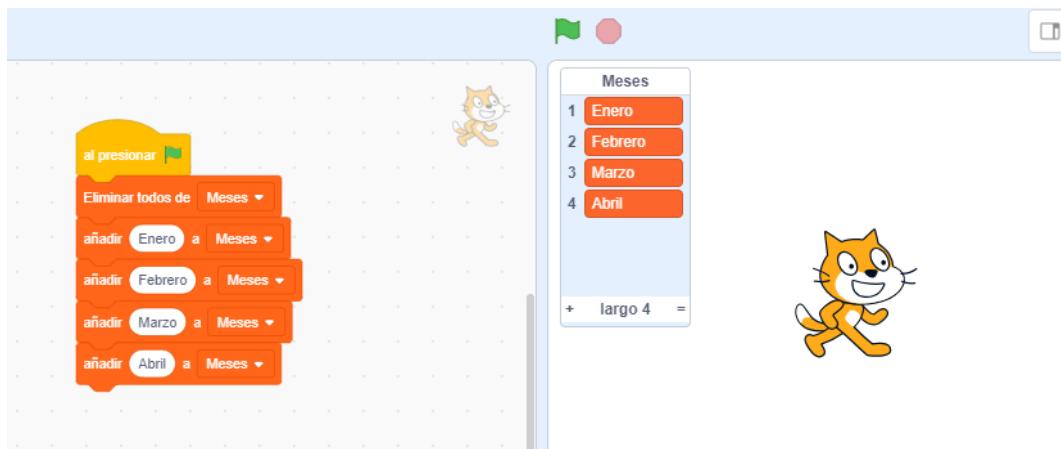


Meses	
1	Enero
2	Febrero
3	Marzo
4	Abril
+ largo 4 =	



Nos quedaría de la siguiente manera

Colocar el bloque eliminar para que no se repita los nombre cada vez que presionamos la bandera verde

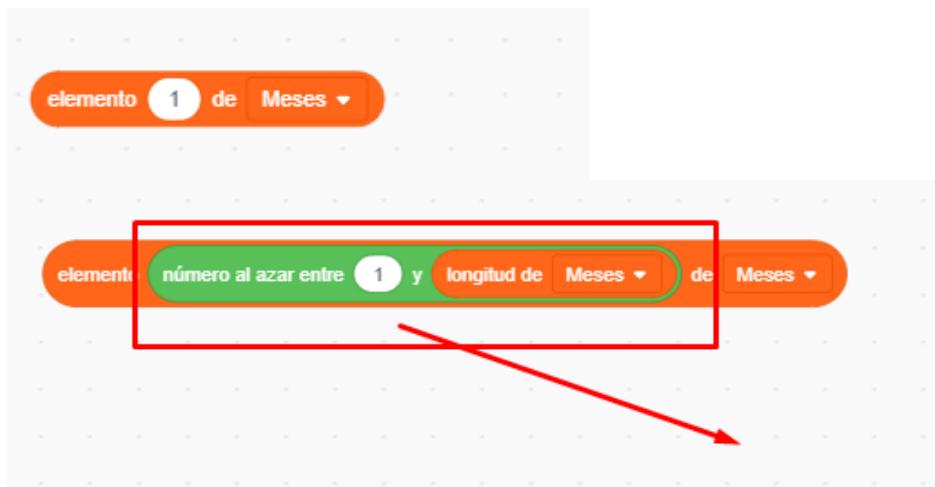


Agreguemos una acción a nuestro Gato



Que nos indique aleatoriamente un mes.

## Bloque Aleatorio



Entre 1 y la longitud de nuestra lista meses

Probemos este código



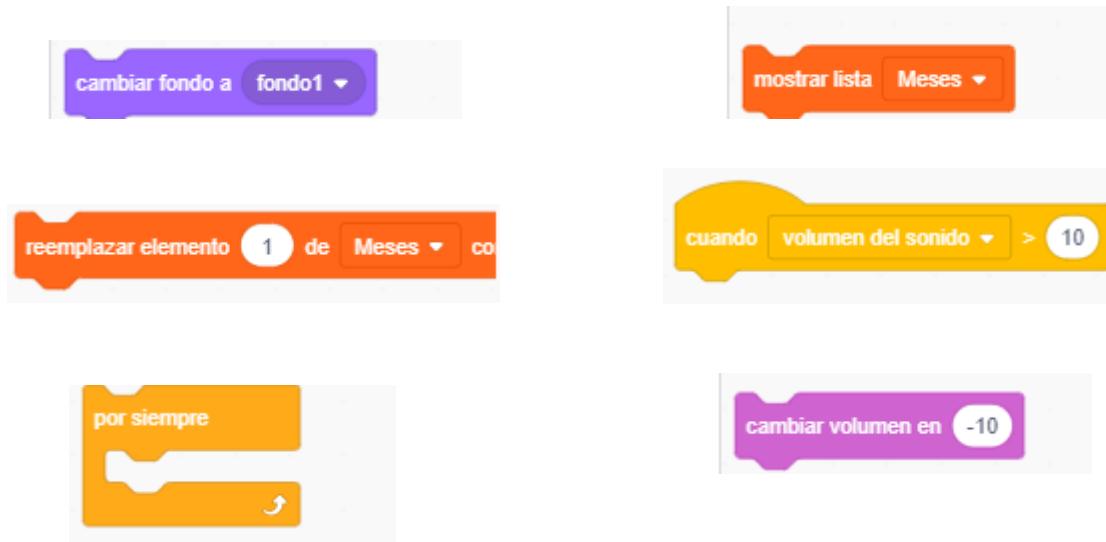
Fecha:  
Nombre:

## TALLER UNIDAD 4



### SCRATCH

1. Encierre las siguientes imágenes de color azul los operadores de listas:



2. Mencione los tipos de Lista.

3. Coloca la letra (M) si pertenece Modo de visualización de variable.

Deslizador

Insertar

Grande

Normal

Disfraz

Repetir

4. Coloque una X dentro del circulo que tenga el concepto correcto de lista.

Estructura de datos referenciada por un nombre que permite almacenar .....

Pueden ser accedidas desde cualquier objeto del programa.

Estructura de datos No referenciada por un nombre que permite almacenar .....

5. Busque y marque con una X los operadores de la lista.

 Borrar Añadir Remplazar Movimiento Insertar Objeto

## Unidad 5



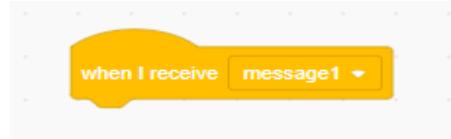
### Eventos

- Un evento ayuda a que múltiples programas en ejecución puedan comunicarse con otros.
- Un evento también permite que se comuniquen los objetos entre sí

Un bloque que invoca un evento es:



Un bloque que recibe un evento es:



Un evento también puede ser generado por acciones del usuario



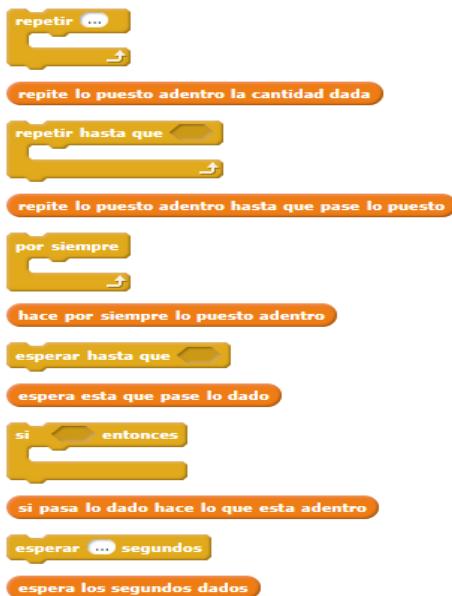
### Eventos de teclado



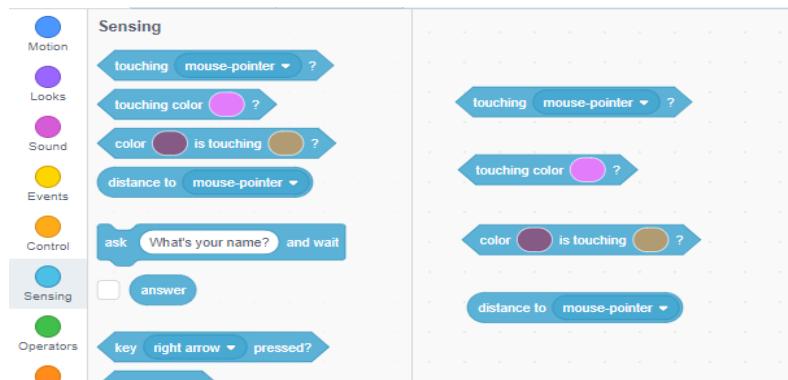
Cuando pulsas una tecla, Scratch se moverá hacia ese lado indefinidamente hasta pulsar otra tecla. Dicho de otro modo, utilizamos el evento de teclado en Scratch para mover al protagonista del juego

## Controles

Las instrucciones que permiten mantener el flujo de un programa en Scratch se encuentran en el Bloque de Control



## Sensores



En Scratch un bloque de sensor permite detectar las siguientes acciones:

- Tocar otro objeto
- Tocar un color determinado
- Cuando un color determinado toca otro
- Entrada de variables por medio de teclado
- Si se presiona una tecla del ratón
- Distancia de nuestro objeto a otro

El bloque de sensores es uno de los bloques realmente importantes, ya que sin este la animación / juego sería algo monótona. aunque siendo unas de los bloques que se complican más en utilizar.

## Animaciones

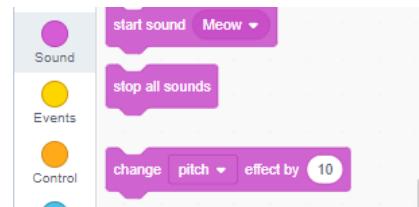
### Movimiento

Contiene instrucciones que permiten mover al personaje o situarlo en cualquier parte del escenario.



### Sonido

Se puede reproducir archivos de sonido dentro de la animación que se está realizando. Esos sonidos se guardan en la pestaña sonido, donde se encuentran las siguientes opciones:

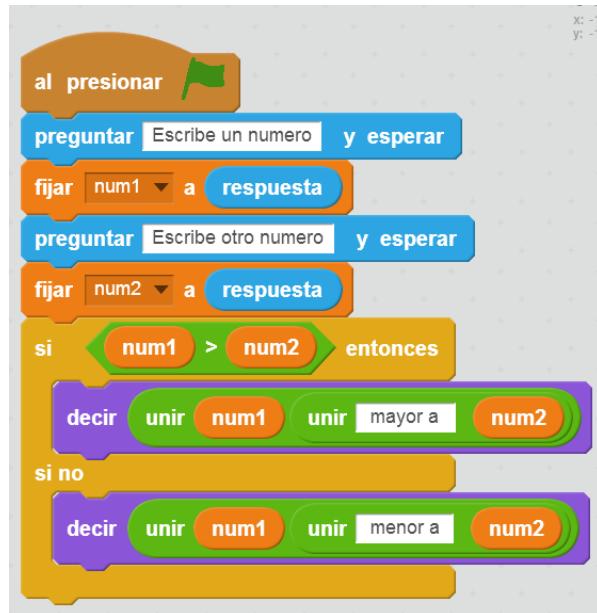


Fecha:  
Nombre:

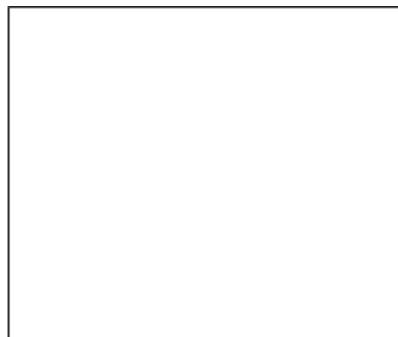
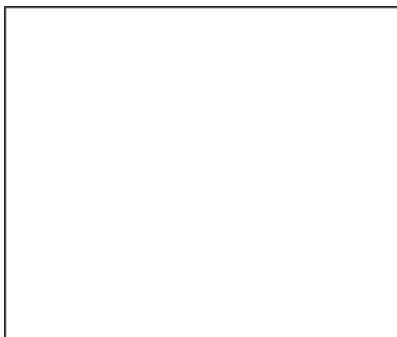
## TALLER UNIDAD 5

### CONTROLES, SENSORES Y ANIMACIONES

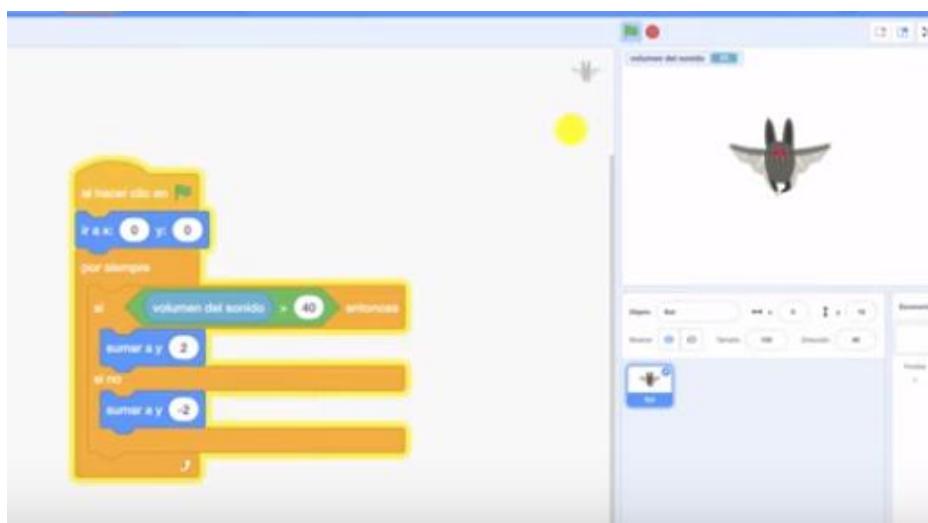
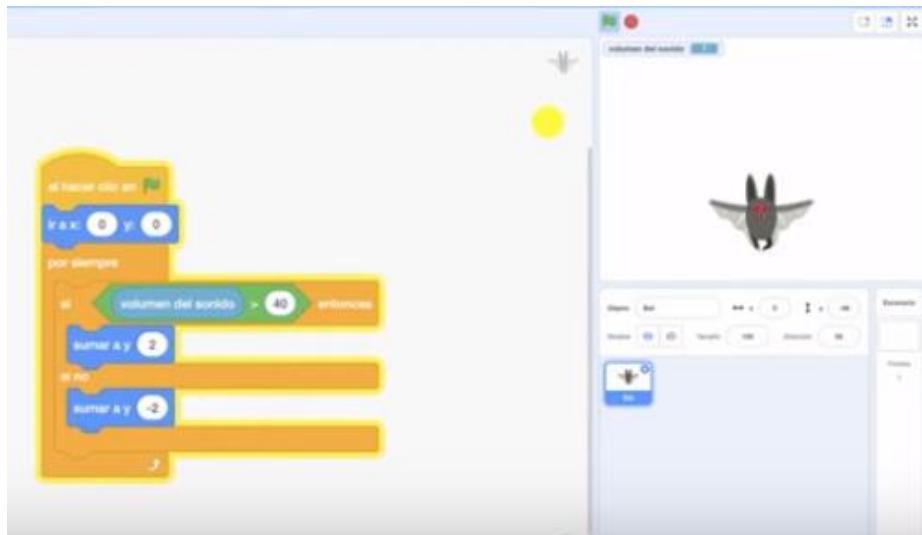
**1. Escriba el flujo de instrucciones del siguiente programa:**



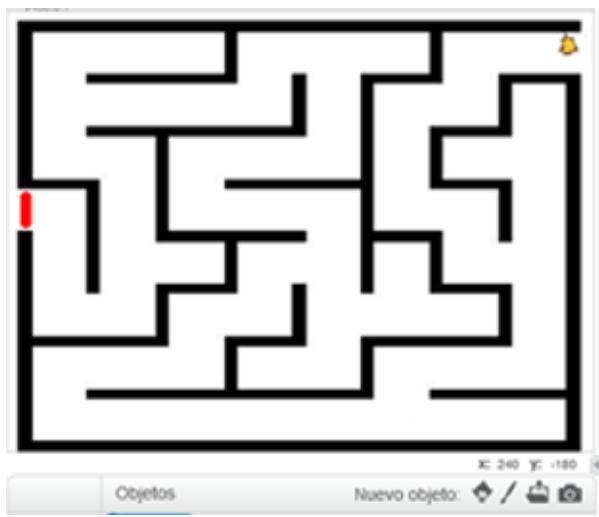
**2. Dibuje 2 etiquetas de operador y 2 etiquetas de sensor en Scratch**



3. ¿Qué realiza el siguiente algoritmo? ¿Que realizan las etiquetas?



4. Al laberinto visto en clase, añadir la funcionalidad de completarlo en 2 minutos acaba el juego.



**GAME  
OVER**

## Unidad 6



### Por qué usar lógica

Si necesito cruzar una calle transitada necesito ayuda de la lógica para saber si no vienen coches en los dos sentidos y así evitar que me atropellen.

Vienen coches por la izquierda “Y” vienen coches por la derecha.

Verdadero	No Cruzar.
Falso	Cruzar.

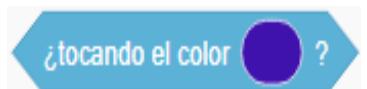
### EXPRESIÓN BOOLEANA

- Las expresiones Booleanas, son expresiones que son o ciertas o falsas.
- Cualquier bloque en forma de diamante alargado es una expresión booleana.
- Un Bloque de este tipo:

### Algunas etiquetas booleanas



Instrucción que realiza una operación booleana para verificar si el puntero del mouse está tocando nuestro objeto.



Instrucción que realiza una operación booleana para verificar si nuestro objeto está tocando el color azul.



Instrucción que realiza una operación booleana para verificar si el color celeste está tocando al color verde.



Instrucción que realiza una operación booleana para verificar si la tecla “espacio” está presionada.



Instrucción que realiza una operación booleana para verificar que dos expresiones sean verdaderas.



Instrucción que realiza una operación booleana lógica para realizar una inversión del valor obtenido de la expresión booleana, convierte un verdadero a falso y un falso a verdadero.



Instrucción que realiza una operación booleana para verificar si alguna de las dos expresiones es verdadera.



Instrucción que realiza una operación booleana para verificar que una expresión contenga otra expresión.



Instrucción que realiza una operación booleana para verificar que dos números son iguales.

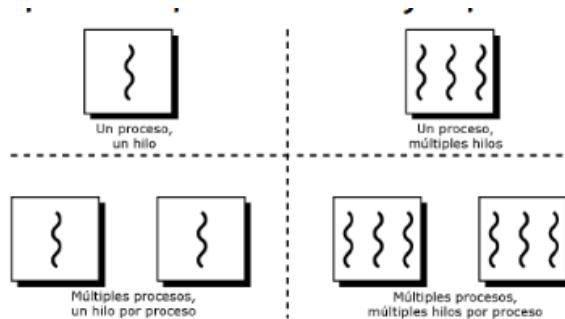


Instrucción que realiza una operación relacional booleana para determinar cuál de los dos números es mayor



Instrucción que realiza una operación relacional booleana para determinar cuál de los dos números es menor.

## Hilos



Los hilos, son básicamente, pequeños bloques de procesos o piezas independientes de un gran proceso.

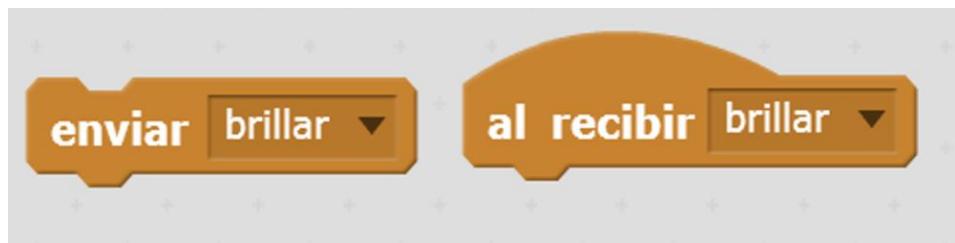
Poner en marcha dos pilas de instrucciones al mismo tiempo hace que se creen dos hilos independientes que se ejecutan en paralelo.

## Multihilos

- Es como un mini código dentro de un programa que se pueda ejecutar al mismo tiempo que lo hacen otros hilos, entonces, un programa con multihilos puede hacer muchas cosas al mismo tiempo.
- Cualquier bloque cuya etiqueta comience con "al presionar", está indicando el inicio de un hilo.



## Eventos



- Los hilos pueden comunicarse unos con otros mediante invocación y manejo de eventos.
- Un evento, entonces es como un mensaje que un hilo le envía al otro.
- Los bloques cuyas etiquetas comienzan con "enviar" invocan eventos, así como los bloques con etiqueta que comienzan con "al recibir" manejan eventos.
- Un evento no solo puede ser invocado por un bloque, sino que las acciones de un usuario pueden invocarlo también.
- Por ejemplo, al hacer clic en la bandera verde de Scratch, efectivamente se invoca un evento que se maneja por:

## Aleatoriedad.



Hay ocasiones en las que necesitamos que nuestros personajes se comporten de manera aleatoria, al azar. A veces de un modo, a veces de otro

Dentro de la categoría de Operadores de Scratch encontramos una instrucción que permite simular un comportamiento muy habitual en el mundo de la programación: la generación de números aleatorios

**número al azar entre 1 y 10**

Una de las claves de utilizar los números aleatorios en Scratch es que, generalmente, hay que acompañarlos con una variable que ‘guardé’ el valor para luego poder jugar con él.



Fecha:  
Nombre:

# TALLER UNIDAD 6



## Lógica Booleana, Hilos

### 1. Escribe (V)Verdadero o (F)Falso según sea el caso:

- a) Las expresiones Booleanas, son expresiones que son o ciertas o falsas. ( )
- b) Cualquier bloque en forma de circulo ovalado es una expresión booleana. ( )
- c) Un Bloque de este tipo: 
- d) Los hilos o threads, son básicamente, pequeños bloques de procesos o piezas independientes de un gran proceso. ( )
- e) Los hilos pueden comunicarse unos con otros mediante invocación y manejo de eventos. ( )
- f) Un evento no solo puede ser invocado por un bloque, sino que las acciones de un usuario pueden invocarlo también. ( )

### 2. Unir con línea.

Instrucción que realiza una operación booleana para verificar que dos expresiones sean verdaderas.



Instrucción que realiza una operación booleana para verificar si alguna de las dos expresiones es verdadera.



Instrucción que realiza una operación booleana para verificar que una expresión contenga otra expresión.



Dentro de la categoría de Operadores está la instrucción que permite generación de números aleatorios.



### 3. Encierre en un círculo la sección del programa donde se muestra una expresión booleana.



4. Indique el resultado del siguiente código.





## Cambiar Color



<http://scratch.mit.edu>

1

SCRATCH

Cree una Tarjeta:



1. Doble la tarjeta por la mitad.

- 1 -



2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

## Cambiar Color



Nuevo objeto: /

al presionar tecla espacio ▾

cambiar efecto color ▾ por 25

cambiar efecto color ▾ por 25



# Mover el Ritmo



<http://scratch.mit.edu>

2

SCRATCH

# Mover el Ritmo



Nuevo objeto:

```

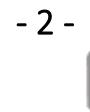
al presionar [bandera]
por siempre
mover (10 pasos)
tocar tambor (1) durante (0.25) pulsos
mover (10 pasos)
tocar tambor (2) durante (0.25) pulsos
  
```



Cree una Tarjeta:



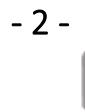
1. Doble la tarjeta por la mitad.



2. Ponga goma en la parte posterior.

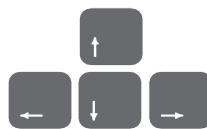
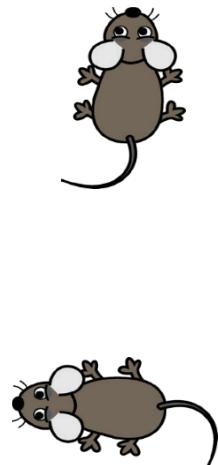


3. Corte a lo largo de la línea punteada.



- 2 -

# Movimientos Glove



<http://scratch.mit.edu>

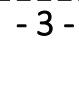
3

SCRATCH

Cree una Tarjeta:



1. Doble la tarjeta por la mitad.



2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

# Movimientos Glove



al presionar tecla flecha arriba ▾

apuntar en dirección 0 ▾

mover 10 pasos

al presionar tecla flecha abajo ▾

apuntar en dirección 180 ▾

mover 10 pasos

al presionar tecla flecha izquierda ▾

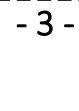
apuntar en dirección -90 ▾

mover 10 pasos

al presionar tecla flecha derecha ▾

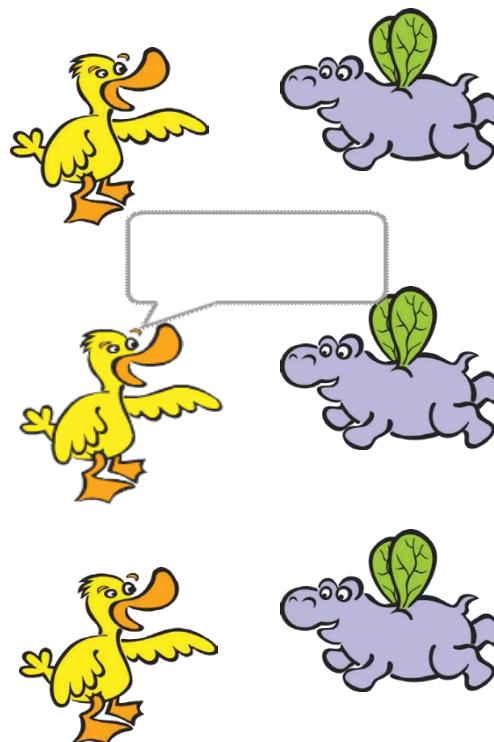
apuntar en dirección 90 ▾

mover 10 pasos



- 3 -

# Diciendo Algo



<http://scratch.mit.edu>

4

SCRATCH

Cree una Tarjeta:



1. Doble la tarjeta por la mitad.

- 4 -



2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

# Diciendo Algo



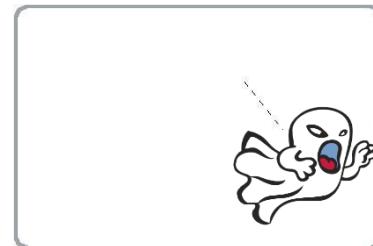
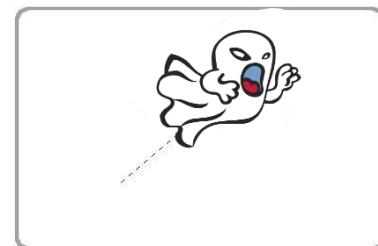
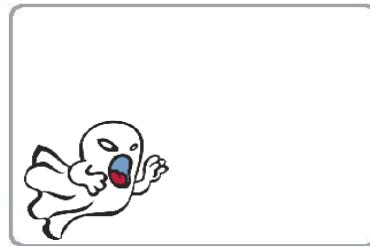
Nuevo objeto:

al clickear este objeto

dicir Hey! No sabía hipopótamos podían volar! por 5 segundos



# Deslizar



<http://scratch.mit.edu>

5

SCRATCH

Cree una Tarjeta:



- Doble la tarjeta por la mitad.

- 5 -



- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

# Deslizar



Nuevo objeto:

al hacer clic en este objeto

deslizar en 1 segs a x: 20 y: 80

deslizar en 1 segs a x: 10 y: -20

deslizar en 2 segs a x: -110 y: -100



i

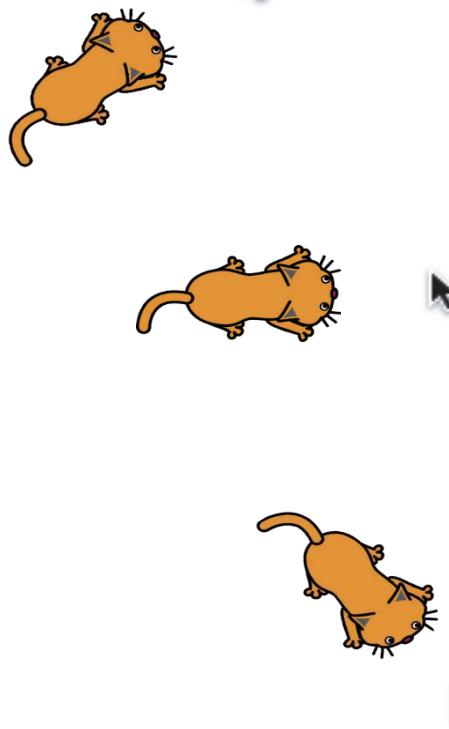
Ghost2  
x: -64 y: -13 dirección: 90°  
estilo de rotación:  
puede ser arrastrado:  
mostrar:

x:-240 y: 180      x: 240 y: 180

x:-240 y:-180      x: 0 y: 0      x: 240 y: -180

x: 0 y: 0

# Sigue al Ratón



<http://scratch.mit.edu>

6

SCRATCH

# Sigue al Ratón



Nuevo objeto:

al hacer clic en este objeto

por siempre

apuntar hacia puntero del ratón ▾

mover (3) pasos



Cree una Tarjeta:



1. Doble la tarjeta por la mitad.



2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

- 6 -

# Bailar Twist



<http://scratch.mit.edu>

7

SCRATCH

Cree una Tarjeta:



- Doble la tarjeta por la mitad.

- 7 -



- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

# Bailar Twist



Nuevo objeto:

Sonido nuevo



```
al presionar tecla [d]
  tocar sonido [human beatbox2 v]
  cambiar efecto [remolino v] por [50]
  esperar [.25] segundos
  cambiar efecto [remolino v] por [0]
  esperar [.25] segundos
```

# Remolino Interactivo



<http://scratch.mit.edu>

8

SCRATCH

# Remolino Interactivo

Nuevo objeto:

```
al presionar  por siempre
establecer efecto remolino ▾ a posición x del ratón
```



x: 240 y: -180  
Nuevo objeto:

Cree una Tarjeta:



- Doble la tarjeta por la mitad.

-8-



- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

# Animélo



<http://scratch.mit.edu>

9

SCRATCH

Cree una Tarjeta:



- Doble la tarjeta por la mitad.

-9-

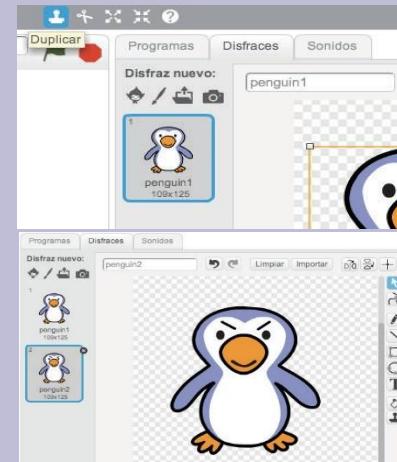


- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

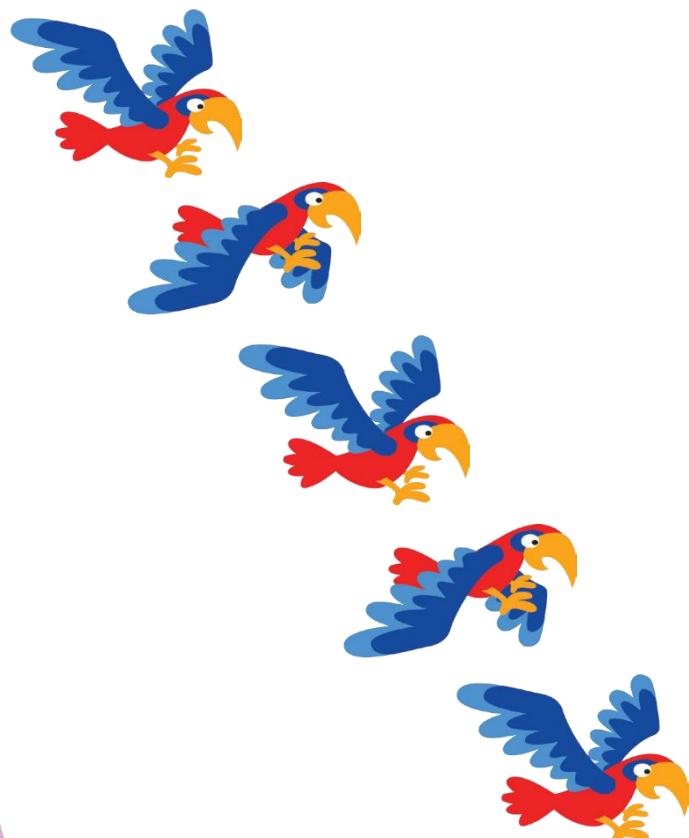
# Animélo



```
al presionar [bandera]
por siempre
  cambiar disfraz a penguin1
  esperar .5 segundos
  cambiar disfraz a penguin2
  esperar .5 segundos
```



## Animación en Movimiento



<http://scratch.mit.edu>

10

SCRATCH

Cree una Tarjeta:

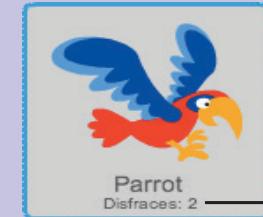


- Doble la tarjeta por la mitad.

## Animación en Movimiento



Nuevo objeto:



Parrot  
Disfraces: 2

```
al presionar 
por siempre
siguiente disfraz
esperar .5 segundos
mover 5 pasos
rebocar si toca un borde
```



Parrot  
x: -41 y: -12 dirección: 90°  
estilo de rotación:  
puede ser arrastrado:  
mostrar:

-10-



- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

# Boton Sorpresa



<http://scratch.mit.edu>

11

SCRATCH

# Boton Sorpresa

Nuevo objeto: ⚡ / 🖌 / 📁 / 📸



Drum1  
x: -73 y: -7 dirección: 90°  
estilo de rotación: ↗ ↔ ⏪  
puede ser arrastrado: ☐

i

al hacer clic en este objeto

cambiar efecto color ▾ por 25

tocar tambor número al azar entre 1 y 18 durante 0.2 pulsos

cambiar efecto color ▾ por -25



Cree una Tarjeta:



1. Doble la tarjeta por la mitad.

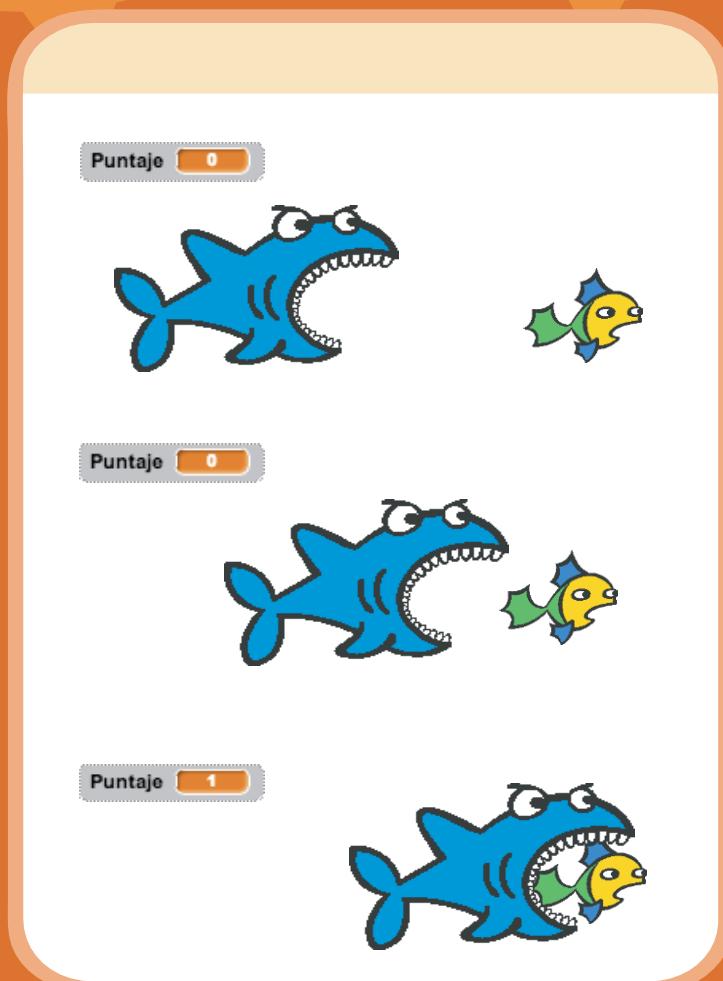


2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

# Guardar Puntaje



<http://scratch.mit.edu>

12

SCRATCH

Cree una Tarjeta:



1. Doble la tarjeta por la mitad.



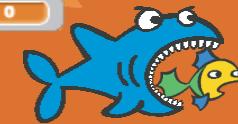
2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

# Guardar Puntaje

Puntaje 0



The image shows the Scratch script for the game and a preview of the printed score card.

**Scratch Script:**

```

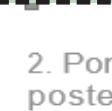
al presionar [bandera]
  fijar Puntaje a 0
  por siempre
    girar (número al azar entre -30 y 30) grados
    mover (5) pasos
    si [¿tocando Fish2?]
      entonces
        cambiar Puntaje por (1)
        tocar sonido [chomp v] y esperar
        mover (-100) pasos
    fin
fin
  
```

**Score Card Preview:**

The preview shows a green card with a white border. At the top left is a green flag icon, and at the top right is a red octagon icon. Below these icons is a dashed line for folding.



1. Doble la tarjeta por la mitad.



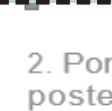
2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.



1. Doble la tarjeta por la mitad.



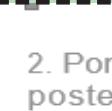
2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.



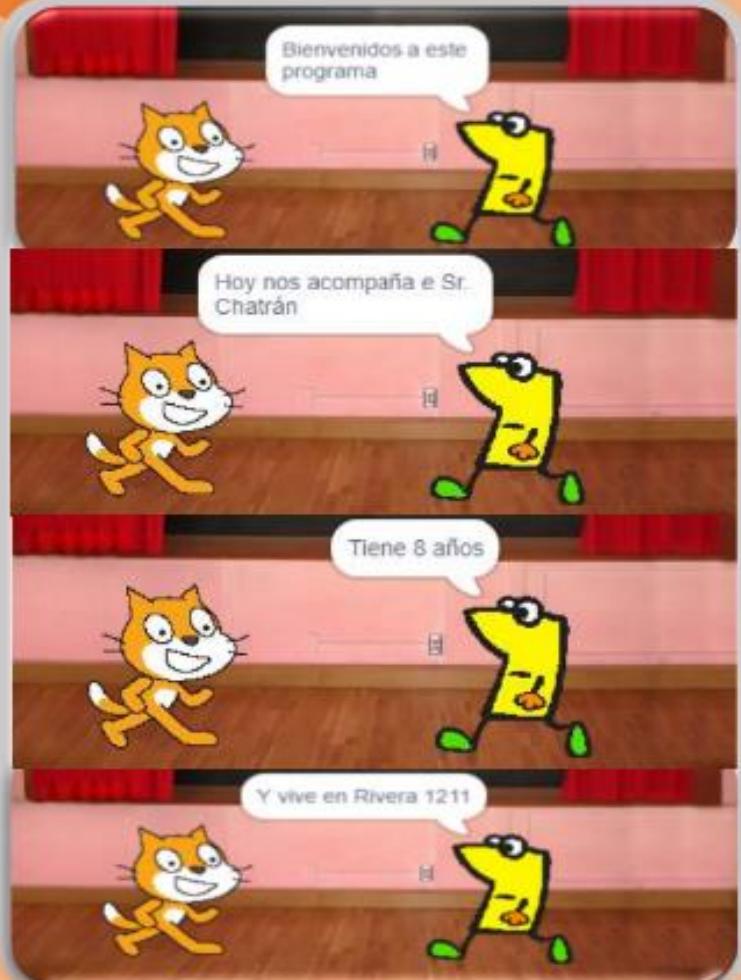
1. Doble la tarjeta por la mitad.



2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.



**PRESENTACION**

**SCRATCH**

**PREPARATE**

Elige **VARIABLES** y haz click en **CREAR UNA VARIABLE**

Pon **nombre gato** como nombre de la variable y haz click en **DE ACUERDO** para crear la variable.

Repite los pasos anteriores y crea 2 variables más con **edad gato** y **dirección gato**

**PRUEBA ESTOS CODIGOS**

```

    al presionar [bandera]
        establecer [nombre gato v] a [Chatrán]
        establecer [edad gato v] a [8]
        establecer [dirección gato v] a [Rivera 1211]
    
```

```

    al presionar [bandera]
        decir [Bienvenidos a este programa] por [2] segundos
        decir [uir Hoy nos acompaña e Sr. nombre gato] por [2] segundos
        decir [uir Tiene v edad gato] años por [2] segundos
        decir [uir Y vive en dirección gato] por [2] segundos
    
```

**HAZLO**

Da **CLICK** en la bandera verde para ejecutar

Cree una Tarjeta:



- Doble la tarjeta por la mitad.

- 13 -

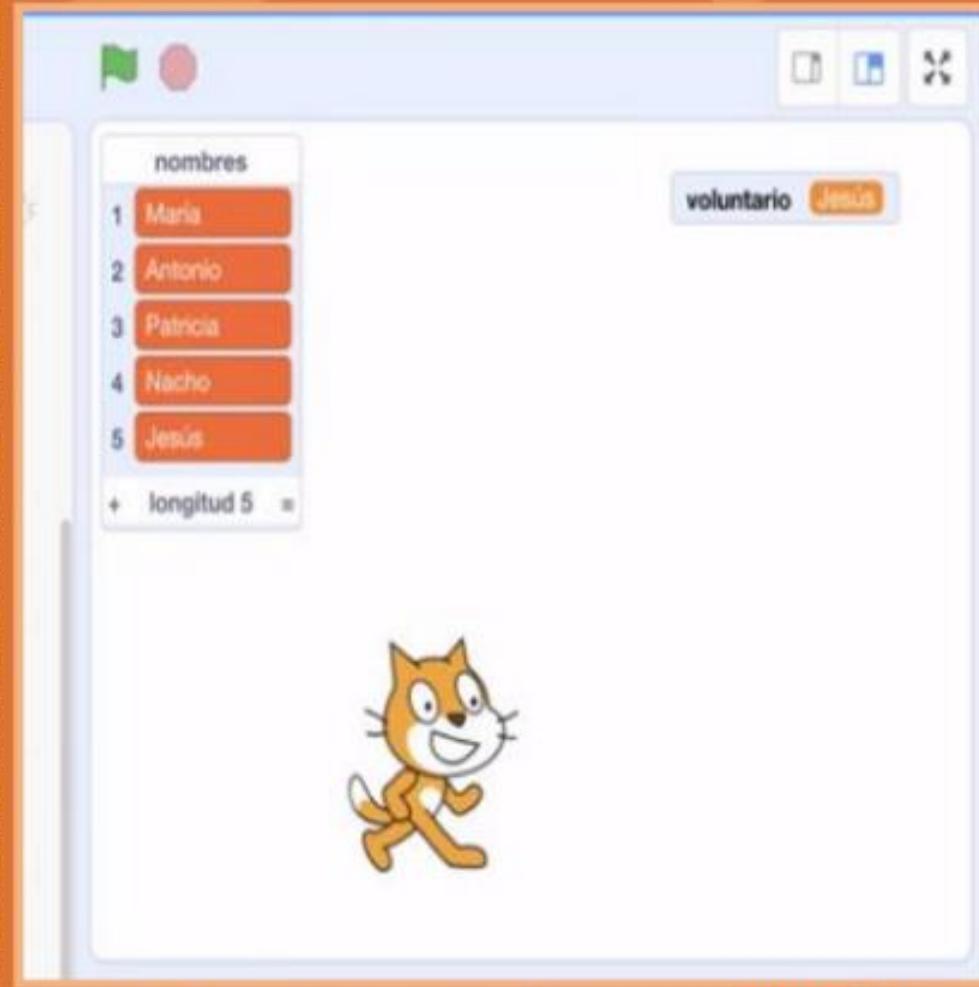


- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

## Crea una lista de voluntarios



**Lista nueva**

Nombre para nueva lista:  
**nombres**

**Prueba este Código**

The Scratch interface shows a new list named "nombres" containing the same five names as the previous list. Below it is a script consisting of several green "repeat" loops. The script starts with a green "repeat" loop with a condition "longitud > 5". Inside the loop, there is a green "repeat" loop with a condition "longitud > 1". Inside this inner loop, there is a green "repeat" loop with a condition "longitud > 0". Inside the innermost loop, there is a green "repeat" loop with a condition "longitud > 0". The script ends with a green "repeat" loop with a condition "longitud > 0".

**HÁZLO!**

Haz clic en la bandera verde para comenzar.

Cree una Tarjeta:



1. Doble la tarjeta por la mitad.



2. Ponga goma en la parte posterior.



3. Corte a lo largo de la línea punteada.

### POZO MILLONARIO



¡Hola! Estos son los números del POZO MILLONARIO

**NúmerosPozo**  
(vacío)

+ largo 0 =

	NúmerosPozo
1	15
2	21
3	17
4	4
5	11
6	19
7	20
+	largo 15

NumeroAzar 5



### POZO MILLONARIO

PREPARATE

Elige **VARIABLES** y haz click en **CREAR UNA VARIABLE**

Pon **NumeroAzar** como nombre de la variable.

También crea una lista **NumerosPozo**.

PRUEBA ESTOS CODIGOS

```

al presionar [espacio]
  Eliminar todos de [NúmerosPozo v-]
  ir a x: -175 y: -100
  decir [¡Hola! Estos son los números del POZO MILLONARIO.] por 2 segundos
  repetir (15)
    decir [DameNúmero]
    decir [¡Felicitaciones al GANADOR!] por 2 segundos
  definir [DameNúmero]
    establecer [NúmeroAzar] a [número al azar entre 1 y 25]
    si [NúmeroPozo contiene NúmeroAzar] entonces
      decir [DameNúmero]
    si no
      añadir [NúmeroAzar] a [NúmeroPozo]
  
```

HAZLO

Da **CLICK** en la bandera verde para ejecutar

Cree una Tarjeta:



- Doble la tarjeta por la mitad.



- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.

- 15 -

### DISFRASADO

**SCRATCH**

### DISFRASADO

**PREPARATE**

Elegir un objeto

Cat-a

Cat-b

Elige **UN OBJETO** que tenga varios **DISFRACES**

**PRUEBA ESTOS CODIGOS**

**HAZLO**

Da **CLICK** en la bandera verde para ejecutar

Cree una Tarjeta:



- Doble la tarjeta por la mitad.



- Ponga goma en la parte posterior.



- Corte a lo largo de la línea punteada.