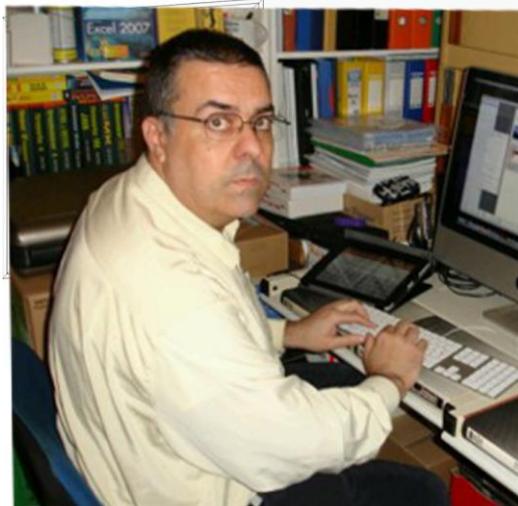


# Introducción a Scratch 3.0

---



Pere Manel Verdugo Zamora  
Correo: [pereverdugo@gmail.com](mailto:pereverdugo@gmail.com)  
Web: [www.peremanelv.com](http://www.peremanelv.com)



# 1.- Programación sin código

## Versiones Scratch

### Aplicación ordenador vs aplicación móvil (Scratch vs Scratch JR)

Scratch Jr es una adaptación de Scratch para que los mas pequeños (5 a 7 años) puedan aprender a programar jugando. Está, por tanto, especialmente pensada y diseñada para niños no lectores o que están empezando a leer. Está en versión IOS y Android.

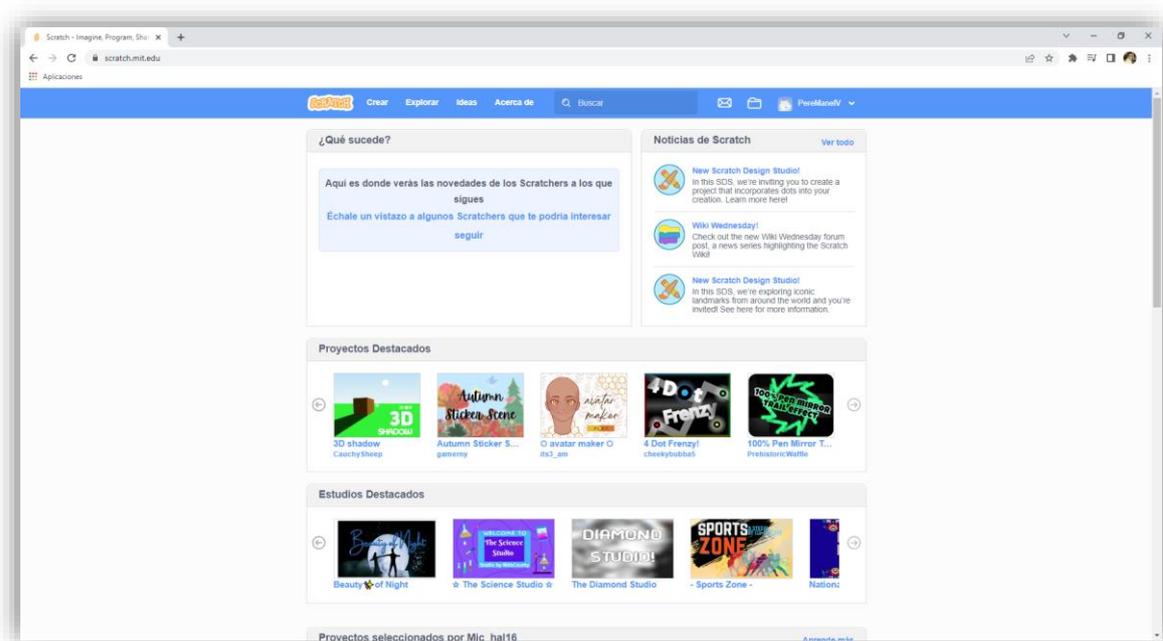


### Aplicación web (online) vs Instalación

- **Aplicación Web (Online).** No hace falta nada y con una cuenta puedo almacenar/compartir mis trabajos en la nube, solo necesitamos un navegador.
- **Aplicación instalable (Offline):** MAC, Linux, Windows y Chromebook.



Para ello vamos a acceder al siguiente enlace: <https://scratch.mit.edu/>



Se puede descargar para los siguientes sistemas operativos desde el siguiente enlace:

<https://scratch.mit.edu/download>



La descargamos e la instalamos en nuestro ordenador, de este modo podríamos trabajar sin conexión a internet, pero ya no trabajaríamos en la nube y podríamos compartir nuestros proyectos si trabajamos desde la versión online.

#### Versiones:

- Versión 1.0. Lanzada en 2007. No tiene ni clones.
- Versión 2.0. Anterior.
- Versión 3.0. Versión actual. Se apoya en HTML5, con una nueva tecnología de bloques (Scratch Block, basada en la tecnología de bloques de Google Blockly), que permitirá mayor conectividad con otras plataformas/desarrolladores externos (Lego, etc).



#### Acceso a la Plataforma

- Acceso a la plataforma Web:

<https://scratch.mit.edu>

- Registro en la plataforma Web:
  - Almacenar nuestras aplicaciones en Internet y por tanto poder acceder desde cualquier lugar.

- Poder compartir mis aplicaciones y que las vea cualquier persona desde un navegador Web y además que pueda ver el código con el que ha sido desarrollado para descargarlo y poder reutilizarlo, modificarlo, etc.
- Importante verificar la cuenta mediante un Email (para poder compartir proyectos).

Es importante que os deis de alta para ello desde la parte superior:



Seleccionaremos “Únete a Scratch”.

The image shows the 'Únete a Scratch' registration form. At the top, it says 'Únete a Scratch' and 'Crea proyectos, comparte ideas, haz amistades. ¡Es gratuito!'. Below this, there are three input fields: 'Crea tu nombre de usuario' with the text 'informenat', 'Crear una contraseña' with two rows of dots, and a checkbox labeled 'Mostrar contraseña'. At the bottom, there is an orange button labeled 'Siguiete'.

Seleccionaremos “Siguiete”.

The image shows the '¿En cuál país vives?' form. It features a decorative header with a globe and stars. Below the header, there is a question '¿En cuál país vives?' and a dropdown menu with 'Spain' selected. At the bottom, there is an orange button labeled 'Siguiete'.

Seleccionamos nuestro país seguido de “Siguiete”.

¿Dónde naciste?

Julio 1970

Nosotros mantendremos esta información en privado. ?

Siguiete

Ponemos el mes y el año de nacimiento, seguido de “Siguiete”.

¿Cuál es tu género?

Scratch recibe a gente de todo género.

Femenino

Masculino

No binario

Otro género:

Prefiero no decir

Nosotros mantendremos esta información en privado. ?

Siguiete

Seleccionamos nuestro género, seguido del botón “Siguiete”.

¿Cuál es tu correo electrónico?

Tu\_Correo@electronico.com

Nosotros mantendremos esta información en privado. ?

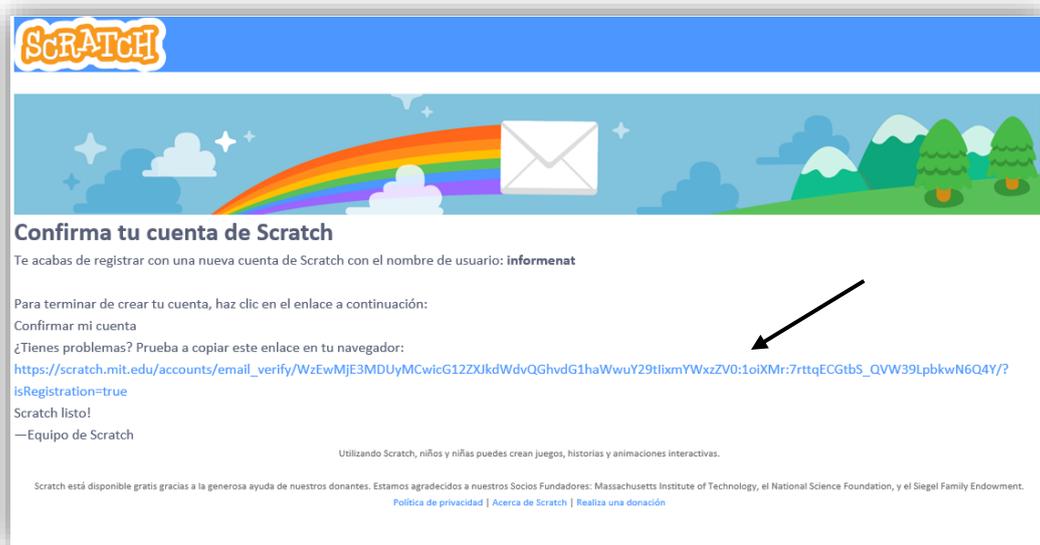
Al crear una cuenta, tu reconoces la [Política de privacidad](#) y aceptas y estás de acuerdo con los [Términos de Uso](#).

Crea tu cuenta

Escribe tu correo electrónico seguido del botón “Crear cuenta”.



Pero antes de empezar vamos a activar nuestra cuenta, para ello vamos a recibir un correo electrónico donde tendremos que activar un enlace.



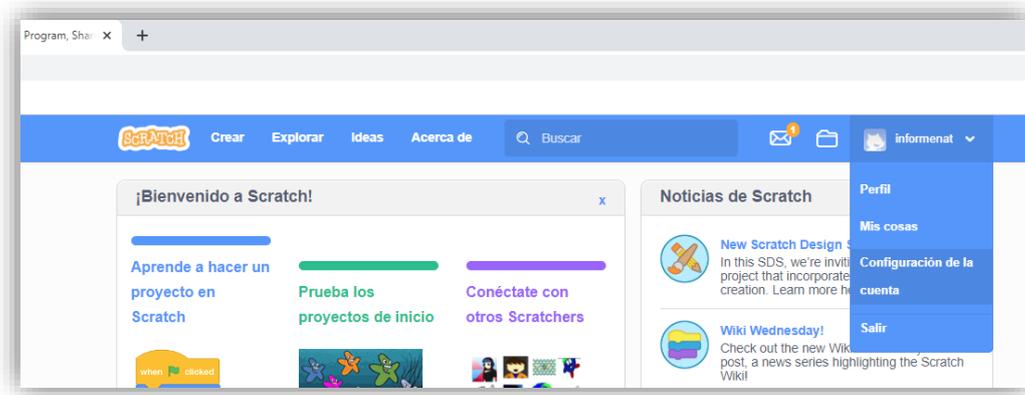
Obtendremos la siguiente respuesta:



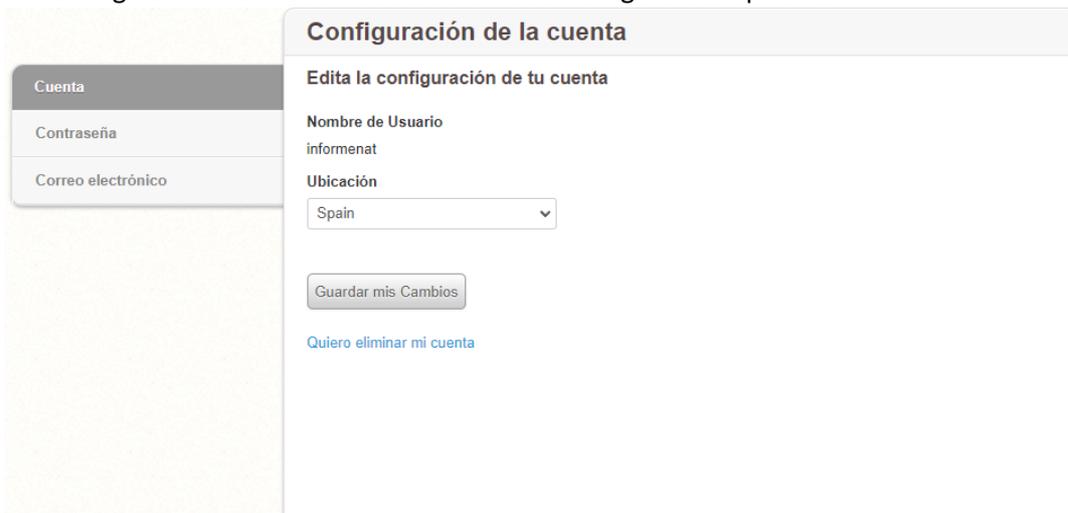
Ya podemos empezar seleccionando Crear.



Cuando tengáis que acceder a vuestra sesión desde otro ordenador, recordar que se accede con el nombre de usuario y la contraseña, tenerla escrita en un lugar seguro por si en un futuro os pudieras olvidar de ella.



Desde configuración de la cuenta accederemos a las siguientes opciones:



En cuenta podemos cambiar el país.



Desde Contraseña podemos cambiar la contraseña poniendo primero la contraseña actual y dos veces la contraseña nueva.

Configuración de la cuenta

Change your email address

Correo electrónico Actual  
pmverdugo@hotmail.com ✓

Type new Email:  
informenat

Confirm password:  
.....

Cambiar Correo Electrónico

Me gustaría recibir novedades por correo electrónico sobre:

- Ideas de actividades para usar Scratch en casa
- Actualizaciones de producto para usar Scratch en un entorno educativo

Desde correo electrónico escribir un nuevo correo electrónico.

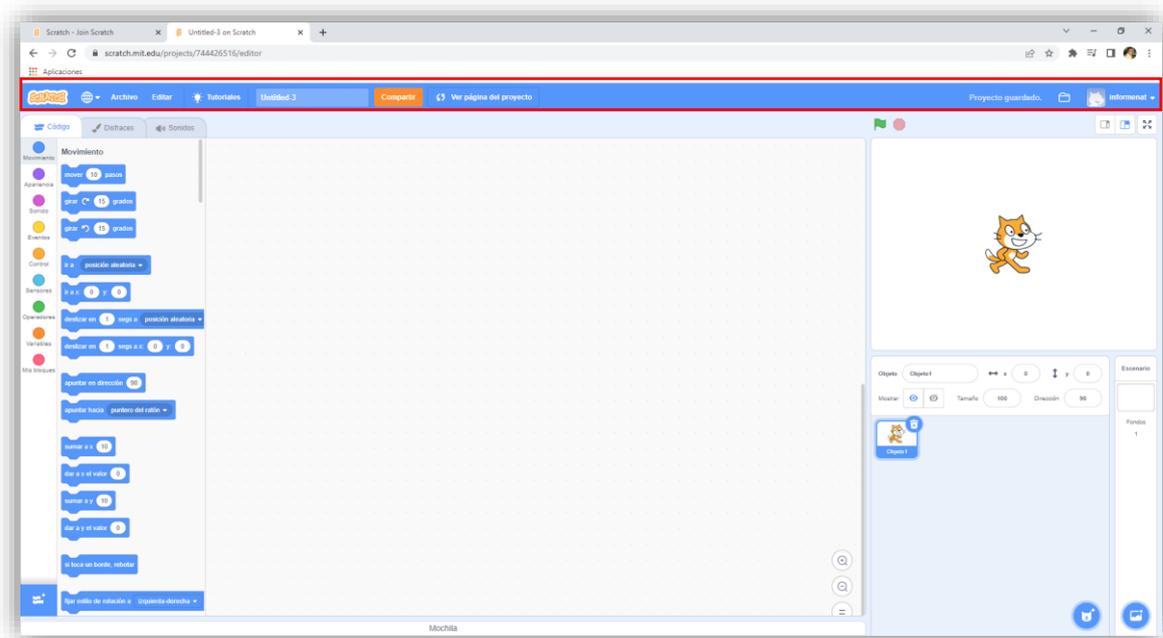
## Interface de Scratch

A continuación repasamos los elementos más importantes de la pantalla de Creación/programación principal de Scratch (Interface):

- **Menú superior:**
  - Idioma.
  - Archivo: Nuevo, Guardar (nube), Guardar un copia (nube), Cargar desde el ordenador (Importar), Guardar en el ordenador (Exportar).
  - Editar: Restaurar (recupera un proyecto) Modo Turbo (hacer que los programas vayan más rápido, esto es muy útil para acelerar los Proyectos matemáticos).
  - Tutoriales: Acceso a los tutoriales.
  - Compartir proyecto.
  - Ir a la página de proyecto.
  - Guardar y mis programas.
- **Zona de código.** Zona con las listas< de todos los bloques de instrucciones y con el código de nuestro programa.
- **Disfraces.** Configuración y edición de disfraces.
- **Zona de ejecución del programa.** Espacio donde se ejecuta y funciona nuestro programa.
- **Zona de objetos y escenarios.** Aquí damos de alta y configuramos los diferentes objetos y escenarios que forma parte de nuestro programa.
- **Mochila.** Zona inferior donde podemos copiar instrucciones de un objeto par poder asignarlas a otro objeto diferente.

Para acceder seleccionamos la opción Crear.





En la parte superior encontramos el menú.



- 1
- 2
- 3
- 4
- 5
- 6
- 7

1.- Nos vuelve a la pantalla de inicio.

2.- Nos permite cambiar el idioma.

3.-



**Nuevo:** crear un nuevo proyecto.

**Guardar ahora:** Guardamos el proyecto.

**Guardar una copia:** Guardamos una copia del proyecto.

**Cargar desde tu ordenador:** Importamos un archivo a nuestro ordenador.

**Guardar en tu ordenador:** exportamos un proyecto a nuestro ordenador.

4.-

**Restaurar:** restaura nuestro proyecto a un paso anterior.

**Activar el modo Turbo:** La ejecución de instrucciones mucho más rápida.

5.-

**Tutoriales:** Podrás acceder a un conjunto de tutoriales.

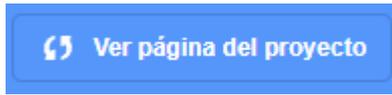
6.-

**Untitled-3:** Dar nombre a nuestro proyecto.

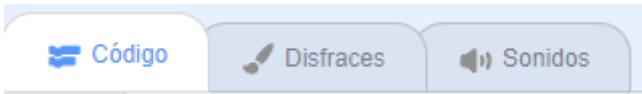
7.-

**Compartir:** Compartir el proyecto.

8.-

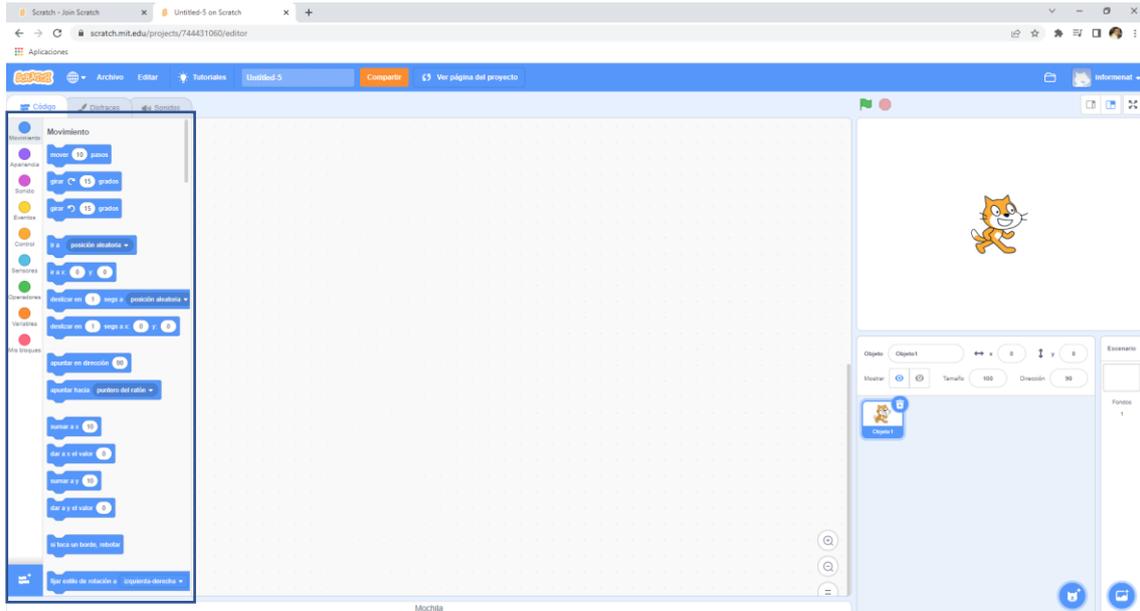


Una vez compartido nuestro proyecto podemos ver la página del proyecto.



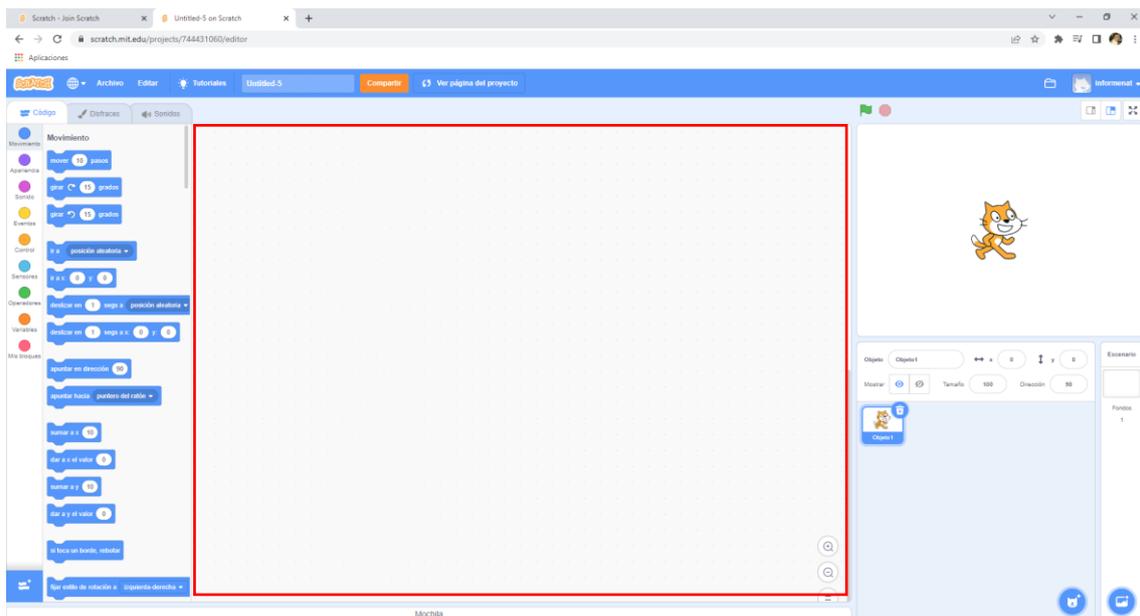
Tenemos la pestaña de Código, Disfraces y Sonidos.

En código:

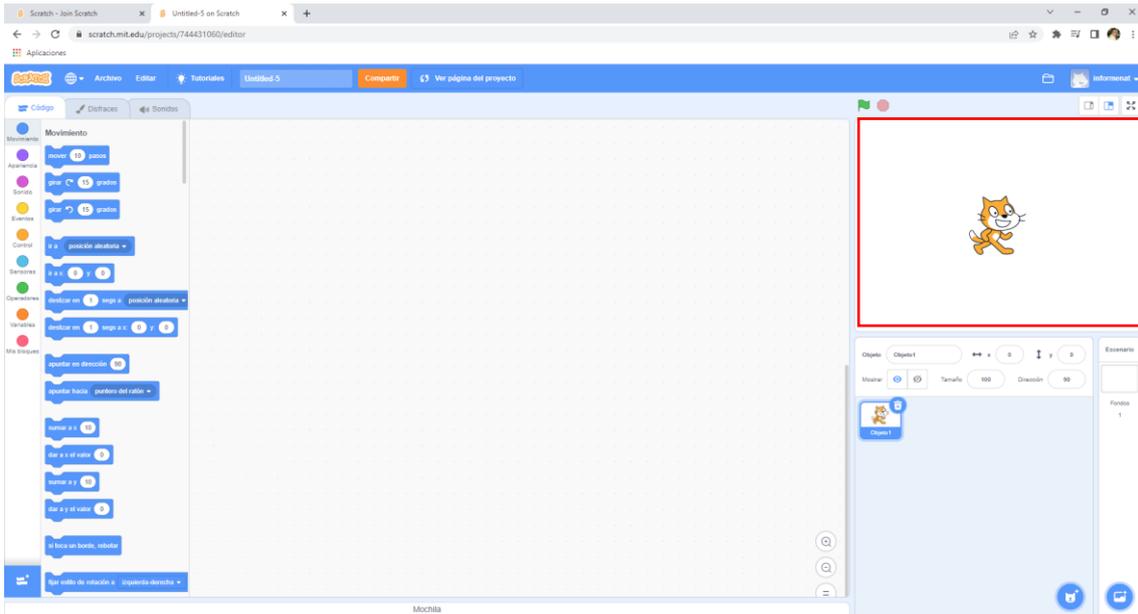


Tendremos acceso a los grupos de bloques y sus respectivos bloques que según el grupo este tendrá un color u otro.

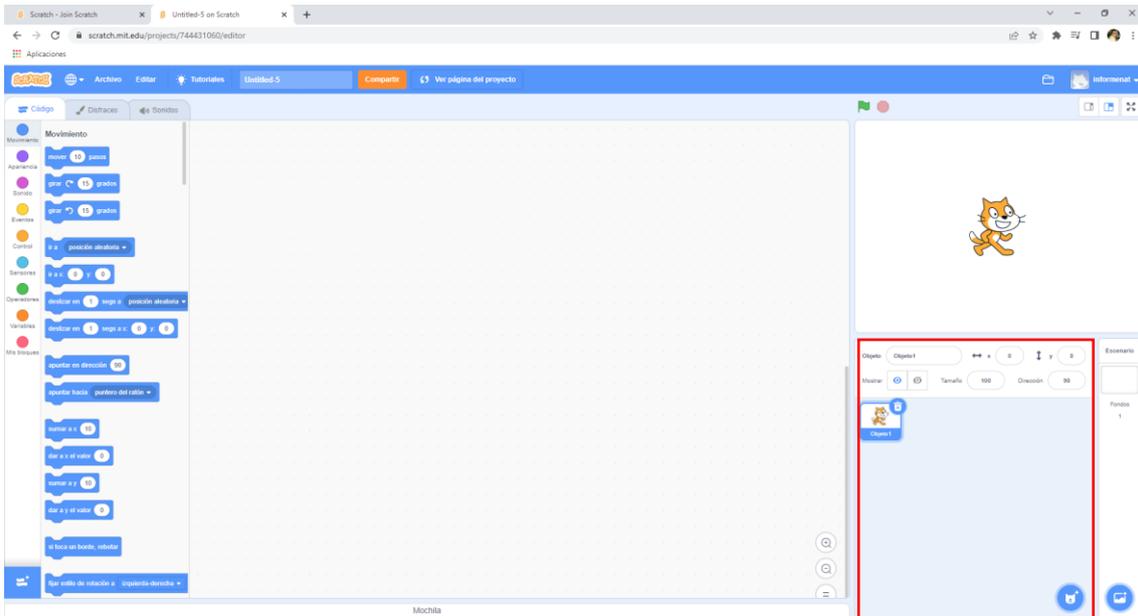
Ventana de programación:



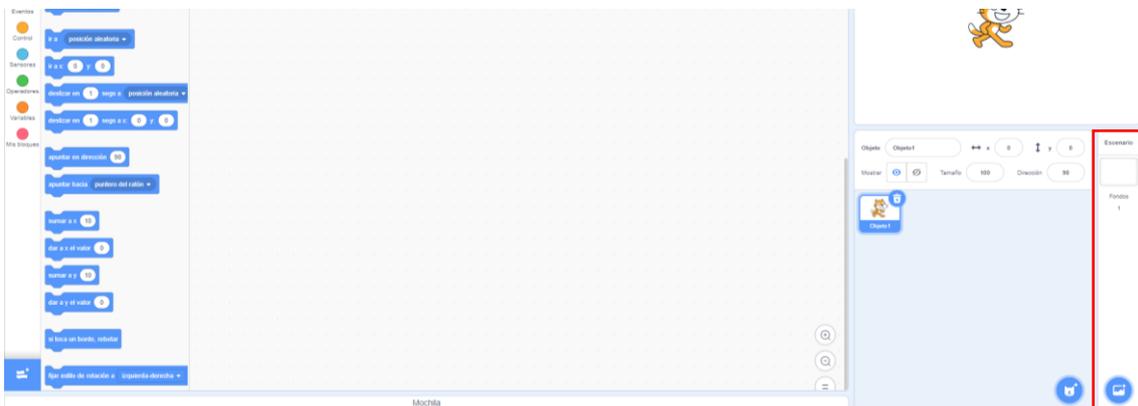
Ventana donde se mostrará el resultado de la programación.



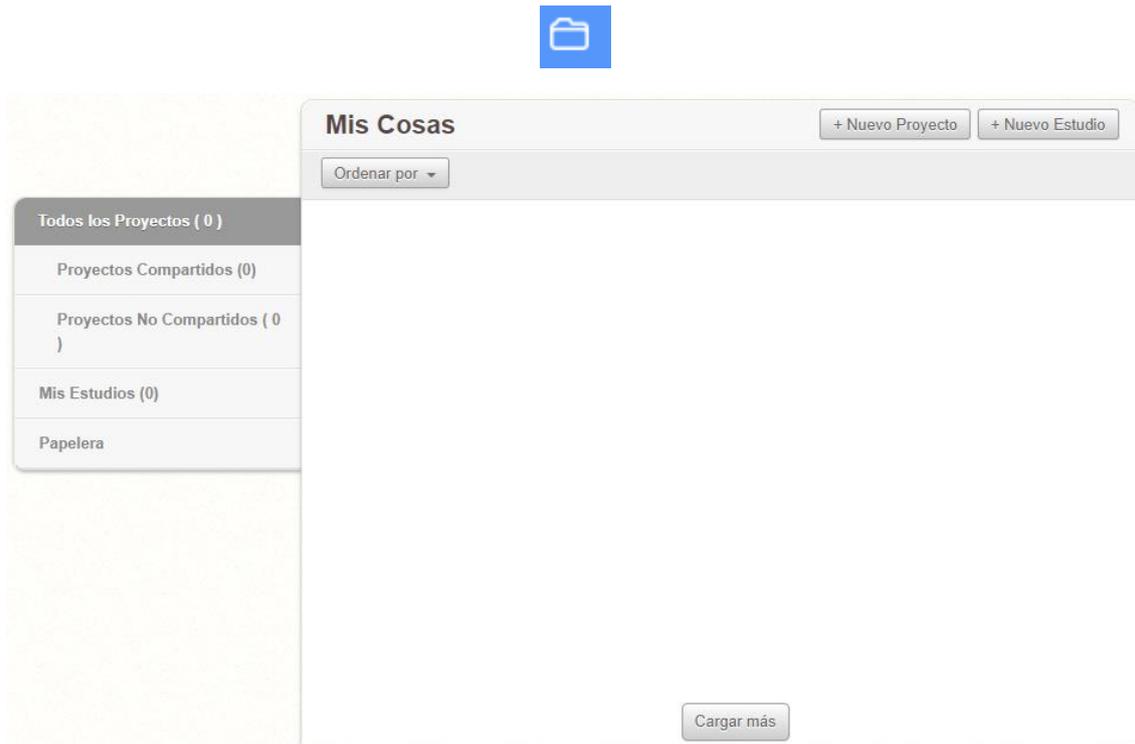
Ventana donde se encuentran los objetos.



Ventana donde se encuentran los escenarios:



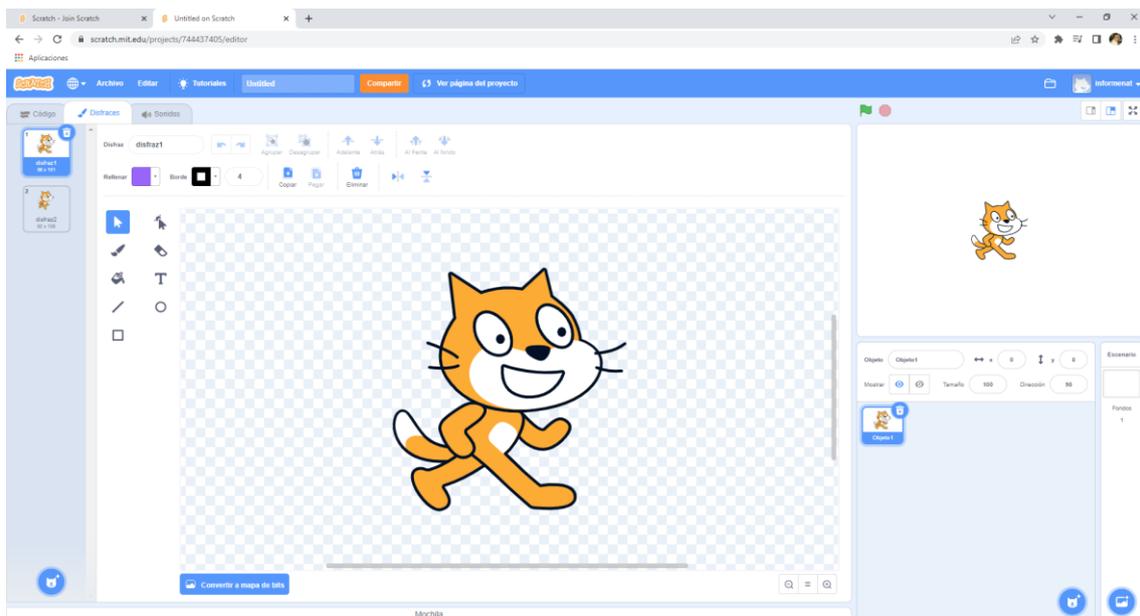
Mi zona:



Donde se pueden ver según la opción que tenemos seleccionada en la parte izquierda:

- Todos los proyectos
- Proyectos compartidos
- Proyectos no compartidos
- Mis estudios
- Papelera

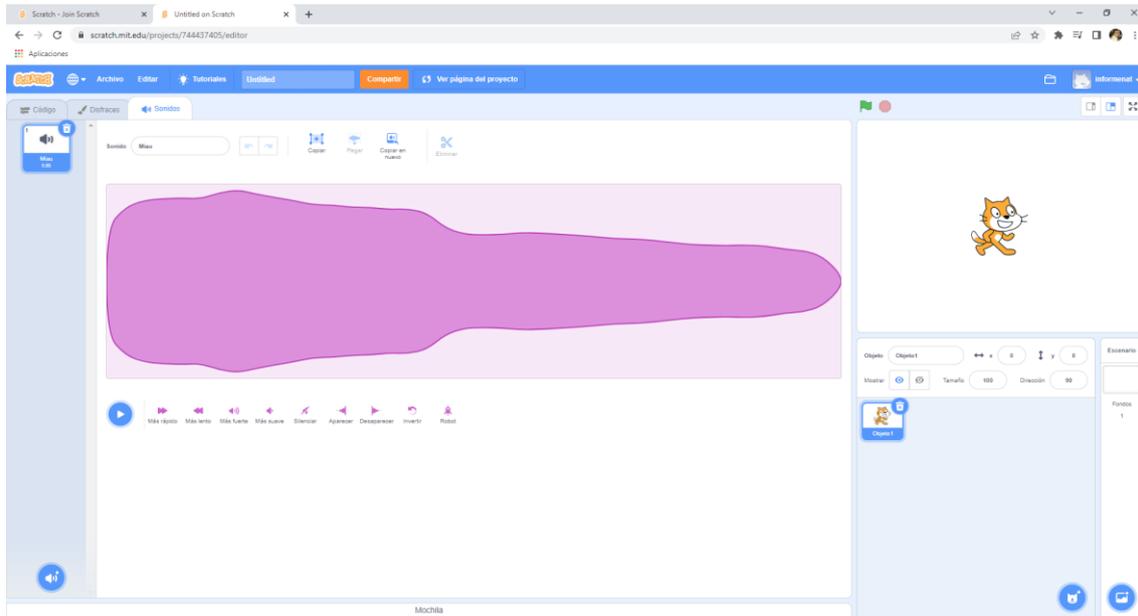
En la pestaña disfraces:



Es donde podemos gestionar los diferentes disfraces por cada uno de mis objetos.

Si estamos seleccionado un fondo esta pestaña se llamara Fondos.

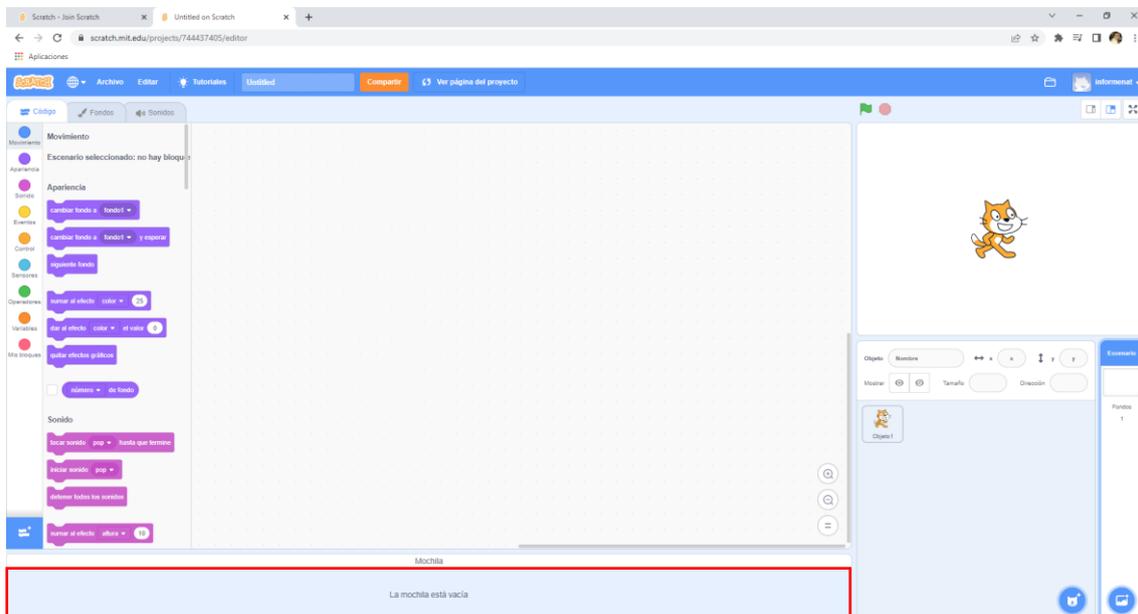
## La pestaña sonido:



Donde podemos modificar sonido, agregar sonidos, etc.

En estos momentos la tengo vacía, la iremos llenando con los proyectos de este tutorial.

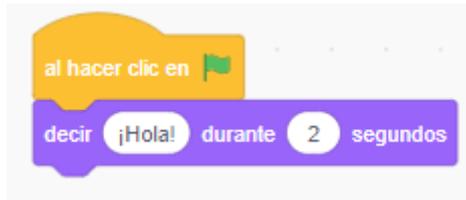
## Mochila:



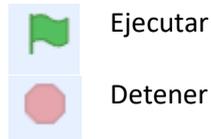
En esta zona podemos arrastrar bloques de código para tener acceso desde otros objetos y así poder reutilizar código.

## Nuestro primer programa en Scratch

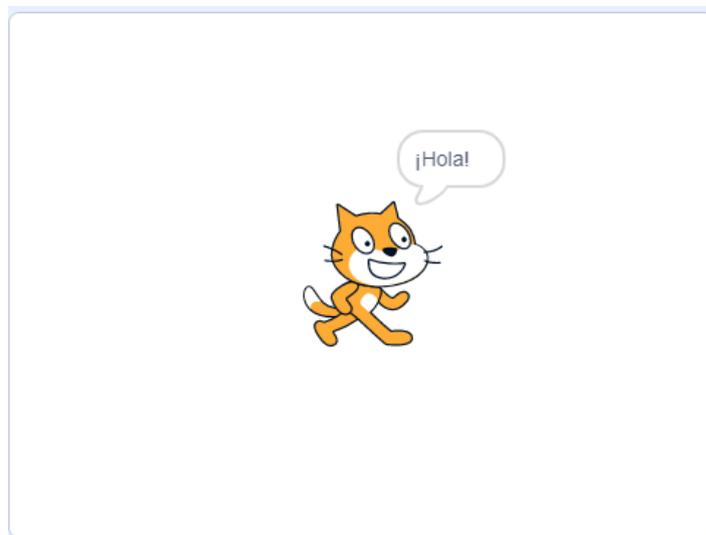
### 1.- Código del programa.



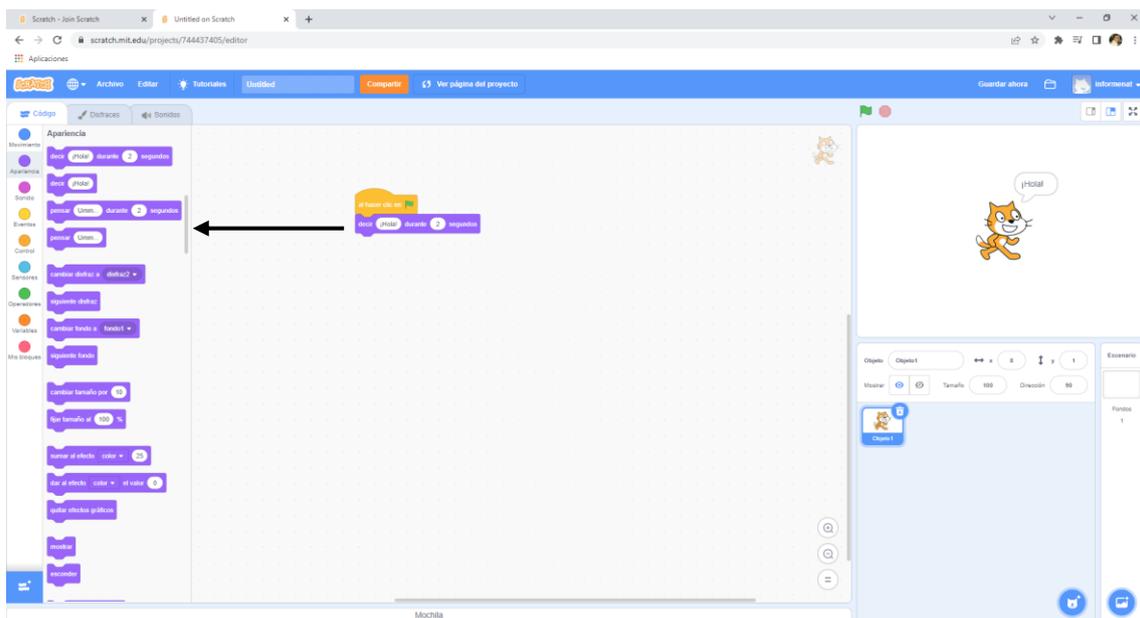
2.- Ejecución del programa.



3.- Este será el resultado.



Cualquier bloque que no deseemos lo seleccionamos y lo arrastramos hacia la izquierda, donde se encuentran todos los bloques.



## 2.- Por qué programar

### ¿Por qué no Programar?

- Está de moda.
- Lo obligan en el colegio.
- Lo demanda el Mercado Laboral y las empresas Tecnológicas.

### ¿Por qué Programar?

1.- Nos da información/conocimiento más CRÍTICO sobre funcionamiento de la Tecnología por dentro, para ser ahora CONSULTORES TECNOLÓGICOS más responsables.

2.- Ahora dejamos de ser meros espectadores a ser CREADORES y formar PARTE ACTIVA DE LA TECNOLOGÍA.

3.- No sólo aprender a programar sino mejor aún PROGRAMAR PARA APRENDER.

- Pensamiento Algorítmico/Computacional. Puede ser la codificación en sí misma no sea el objeto (aunque, por supuesto, necesitamos programadores igual que médicos), pero si la educación de codificación a gran escala ayuda a los estudiantes a desarrollar sus propias capacidades cognitivas, entonces es un medio potencialmente útil para un extremo más grande.
- Pensamiento Creativo.

### Conclusión

- Los ordenadores son increíblemente **rápidos, exactos y estúpidos**.
- Los seres humanos son increíblemente **lentos, inexactos y brillantes**.

“Juntos son inimaginablemente poderosos”

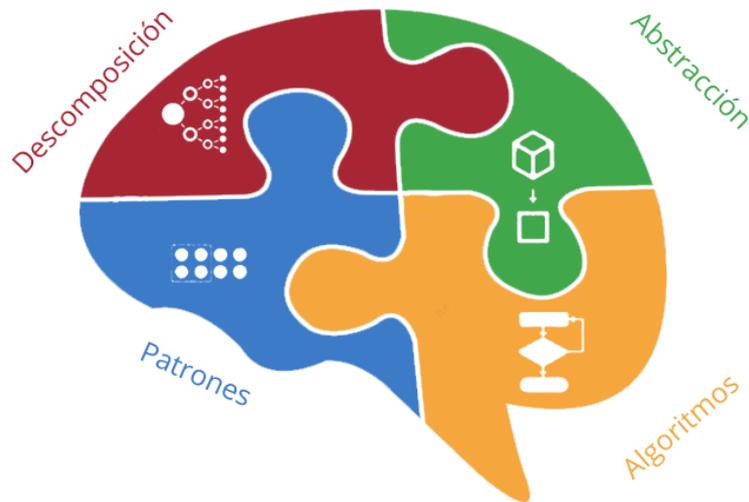
**LOS ORDENADORES SON IMPORTANTES PERO MÁS IMPORTANTE SON AÚN, LAS PERSONAS QUE LOS PROGRAMAN Y EL PROCESO DE PROGRAMACIÓN.**

### 3.- Introducción a la programación

#### Pensamiento computacional

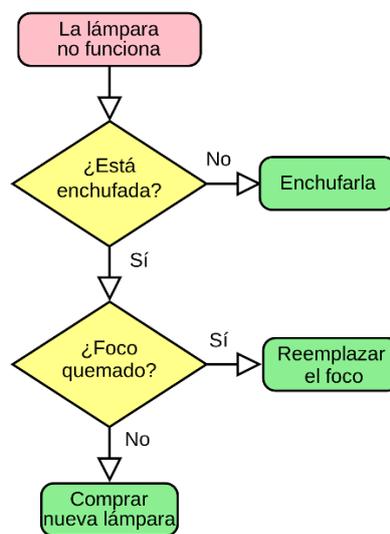
Formular problemas de forma que se permita el uso de un ordenador para ayudar a resolverlos.

Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos.



#### Algorítmicos

Conjunto de instrucciones paso a paso para la solución de una tarea.



#### CARACTERÍSTICAS

- **No ambiguo**, Instrucción sólo puede ser interpretado de una manera única.
- **Ejecutable**. La persona o dispositivo que va a ejecutar la instrucción deben de llevarla a cabo sin ninguna información adicional.
- **Ordenado**. Los pasos de un algoritmo deben ser ordenados en una secuencia adecuada para llevar a cabo correctamente la tarea.
- **Finito**. Debe terminar en algún momento. Número finito de pasos.

## Programa informático

Los dispositivos informáticos son tremendamente potentes pero si no disponen de un programa que les indique lo que tienen que realizar, no son capaces de hacer nada. Un programa informático es:

- Es la TRADUCCIÓN de un algoritmo utilizando un lenguaje que pueda entender el dispositivo informático.
- CONJUNTO DE INSTRUCCIONES desarrolladas en un determinado lenguaje comprensible para un dispositivo informático.
- Como secuencia el dispositivo realiza una serie de acciones.

```
# Programa Elaborado por Cristian Eduardo Leal Campos
# Programa de Elevar al cuadrado

print "Le mostrare el Numero que Desea elevado al cuadrado."
numero = input("Introduzca numero: ") #Guardamos el nombre
print "Su numero es : %s, y el lo vamos a elevar ala potencia 2" % (numero)

final = numero* numero

print final

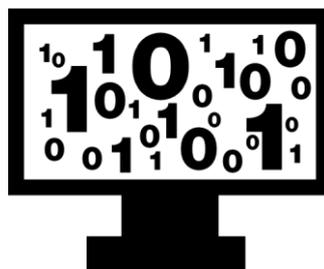
raw_input() #En este caso lo usamos para parar el programa hasta que se pulse enter
```

## Lenguaje de Programación

Para desarrollar un programa es necesario conocer el lenguaje que entiende el ordenador.

### TIPOS DE LENGUAJES

- Si por el contrario el lenguaje utiliza códigos binarios, el lenguaje se denomina de BAJO NIVEL.



- Si el conjunto de instrucciones que forma parte del lenguaje de programación se base en el lenguaje natura humano se denomina de ALTO NIVEL (TEXTO).

```

class ObjetoEjemplo {

    private String propiedad1 = "Soy la propiedad 1";
    private String propiedad2 = "Soy la propiedad 2";
    private int valor1 = 10;

    public String getPropiedad1() {
        return propiedad1;
    }

    public String getPropiedad2() {
        return propiedad2;
    }

    public int getValor1() {
        return valor1;
    }

}

```

- Para hacer aún más fácil iniciamos con la programación unos lenguajes que se programan de forma visual y gráfica a través de bloques. Estos lenguajes se denominan lenguajes de programación VISUALES (BLOQUES).

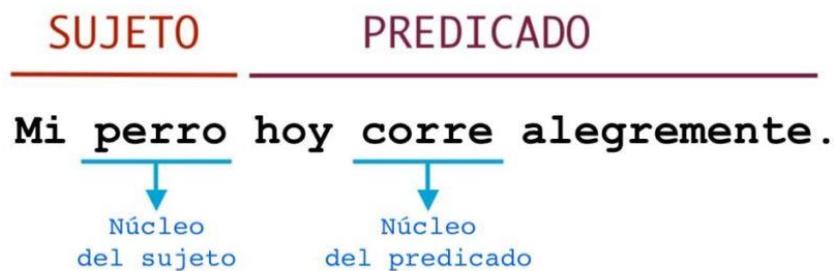


### ELEMENTOS QUE FORMAN PARTE DE UN LENGUAJE DE PROGRAMACIÓN.

- Conjunto de instrucciones de dicho lenguaje. Metáfora de **VOCABULARIO** en un idioma.



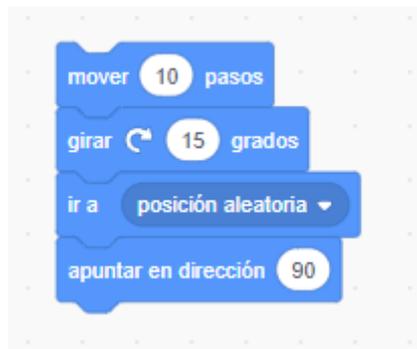
- Las reglas para escribir las instrucciones. Metáfora de **SINTAXIS** en un idioma.



### Elementos básicos de la programación

Aunque cada lenguaje de programación tiene su estilo de programación y su propio conjunto de instrucciones, es cierto que la mayoría comparten una serie de conceptos que de una u otra manera suele aparecer en todos y que supone la BASE DE LA PROGRAMACIÓN. Estos elementos son los siguientes:

- **ESTRUCTURAS BÁSICAS DE PROGRAMACIÓN.** Todos los programas informáticos se construyen en base a la ejecución de una serie de instrucciones que seguirán una de estas 3 posibles secuencias u orden de ejecución.
  1. **ESTRUCTURAS SECUENCIAS.** El programa realiza las acciones secuencialmente, es decir, una instrucción detrás de otra instrucción.



2. **ESTRUCTURAS SELECTIVAS (Condicionales/Selección).** El programa toma un camino u otro en función de una condición.



3. **ESTRUCUTRA REPETITIVAS (Bucle/Repetición).** Repetición de una o varias instrucciones, esta repetición puede estar determinada por:

- Condicionada. La repetición se produce mientras o hasta que se cumpla una condición.



- No Condicionada. La repetición se realiza un número (fijo) determinado de veces.



- **BLOQUES INSTRUCCIONES.** Se suelen denominar funciones o Procedimientos y son un conjunto de instrucciones empaquetadas, para que puedan ser ejecutadas en diversos momentos sin necesidad de escribir todo este conjunto de instrucciones.



- **EVENTOS.** Sucesor que ocurren y que hacen iniciar una nueva línea de ejecución.
  1. Externos. Por parte del usuario: Al pulsar una tecla, ratón.



2. Internos al programa: Cuando queremos comunicar información entre los diferentes objetos del programa.



- **VARIABLES.** Una variable en un programa no es más que un espacio en la memoria del ordenador en el que se almacena un VALOR. (Metafóricamente de cajón de un archivador). Podemos asignar en cualquier momento el VALOR correspondiente en dicha variable (Ejemplo: PUNTOS=10).



- **COMENTARIOS.** Instrucciones que el lenguaje de programación no ejecuta, ignora. Sirven para que los programadores pongamos información aclaratoria indicando como funciona nuestro programa: para recordarlo más adelante o por si otra persona tiene que trabajar con él.



## 4.- Objetos y Escenarios: APARIENCIA

### Interface de nuestro programa en Scratch

Lo primero que tenemos que definir cuando vamos a realizar un programa es el aspecto que este va a presentar al usuario, más adelante añadiremos la programación. Aunque es una practica de programación muy recomendable no perder tiempo inicialmente a dotar a nuestros programas de una apariencia perfecta, recomendamos:

1. Añadir los elementos que vaya a precisar...
2. Realizar entonces la programación.
3. ... y ahora ya cuando nuestro programa funcione, dedicarnos a mejorar el aspecto e interface de nuestro programa.

Scratch trabaja fundamentalmente con dos elementos para el diseño de nuestros programas:

1. **Objetos/Personajes.** Son los diferentes elementos que van a formar parte de nuestro programa y a los que podremos asignar una programación específica a cada uno de ellos.
2. **Fondos para el Escenario.** Diferentes fondos que podremos aplicar en el escenario de nuestro programa, también se pueden programar.

Scratch Jr una adaptación de Scratch para que los más pequeños (5 a 7 años) puedan aprender a programar jugando. Está, por tanto, especialmente pensada y diseñada para niños no lectores o que están empezando a leer. Está en versión iOS y Android.

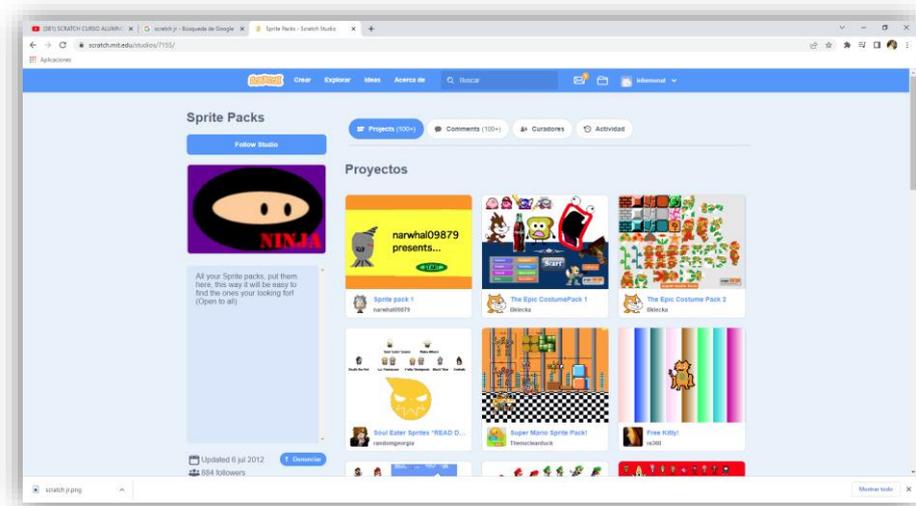


### OBJETOS/PERSONAJES

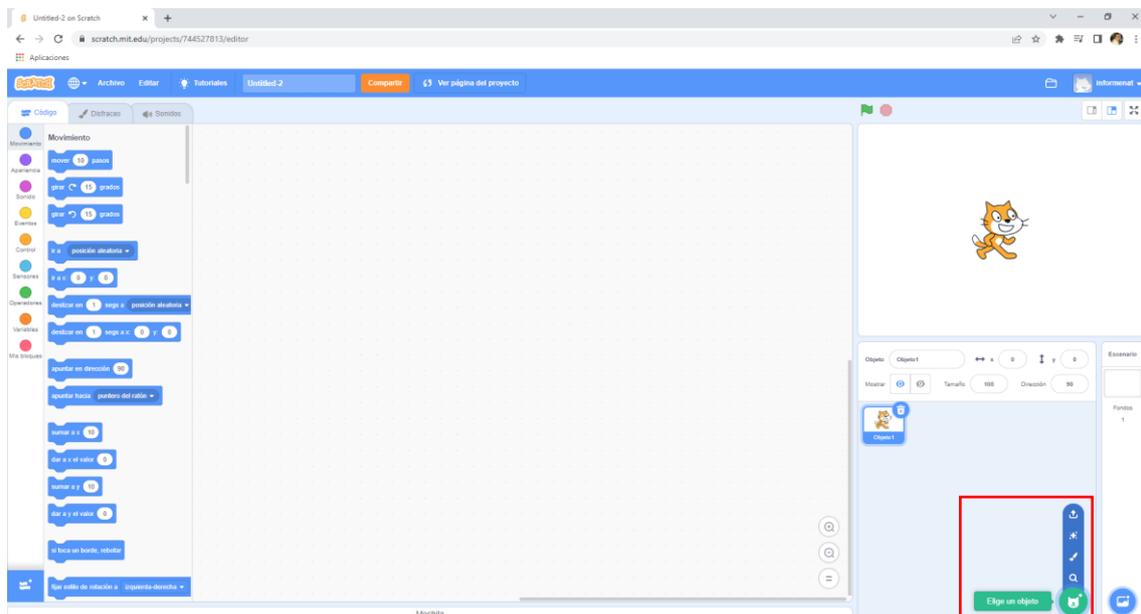
- **CREAR UN NUEVO OBJETO**
  - Subir un Objeto. Puede ser una imagen o un sprite que será un objeto que puede tener diferentes disfraces ya incorporados.
  - Objeto Sorpresa.
  - Dibujar Objeto. Nos permite dibujar nuestro propio objeto en un editor de imágenes.
  - Elegir Objeto. Utilizar uno de los objetos ya incorporados en la biblioteca de Scratch.
- **CONFIGURACIÓN INICIAL OBJETO**
  - Nombre
  - Posición inicial
  - Mostrar/Ocultar
  - Tamaño
  - Dirección
- **DISFRACES. EDITOR DE IMÁGENES.**
  - Cada personaje puede presentar distintos aspectos, en diferentes momentos del programa. Es se denomina DISFRACES en Scratch y

podemos crear tantos como queramos para cada uno de nuestros personajes y luego controlarlos por programación.

- Hay personajes, de la biblioteca de Scratch que ya vienen con varios disfraces incorporados, como por ejemplo el gato de Scratch.
- También podemos incorporar nuestros propios disfraces.
- Finalmente podemos modificar el aspecto de cualquier disfraz utilizando el EDITOR DE IMÁGENES incorporado en Scratch. Recomendamos trabajar en modo VECTORIAL (modo por defecto), para tener una mejor resolución con los objetos.
- SUGERENCIA. Podemos añadir objetos con sus correspondientes disfraces de otros programas de la comunidad, es más, hay páginas (estudios) con un montón de programas dedicadas específicamente a tener personajes para poder exportarlos e importarlos a nuestro programa, os dejamos una: <https://scratch.mit.edu/studios/7155/>



Un sprite es cuando te descargas un objeto de otro proyecto, además de descargarte el objeto también te descargas el programación y todos los disfraces que tiene.



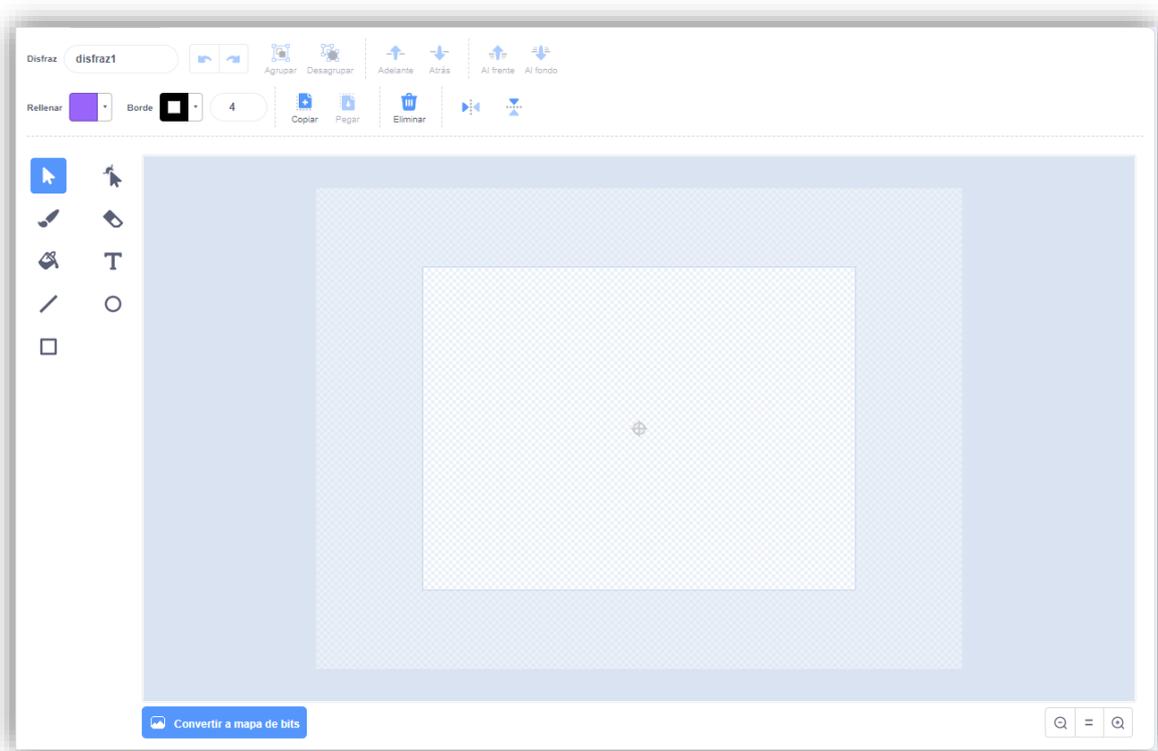


Importar un objeto/spriete desde nuestro ordenador.

Importar un objeto aleatoriamente.

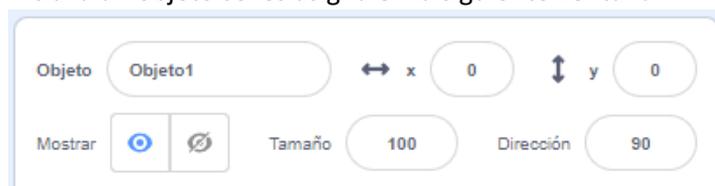
Crear un objeto con el editor de imágenes, siguiente imagen.

Buscar un objeto desde la biblioteca de objetos.



Se puede trabajar en vector y mapa de bits, el más recomendable es el de vector ya que al modificar su tamaño este no perderá calidad, en cambio en mapa de bits si pierde calidad.

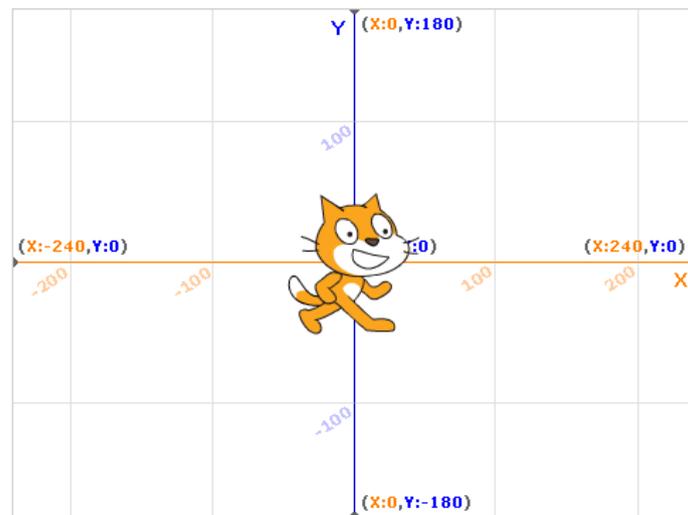
La configuración inicial a un objeto se les asigna en la siguiente ventana:



Nombre del objeto, posición coordenada x, posición coordenada y, mostrar/ocultar objeto, tamaño del objeto y su orientación.

Es aconsejar nombrar bien los objetos ya que si disponemos de varios es una buena forma de poderlos identificar cuando estemos programando.

Con respecto a las coordenadas este son sus valores en el escenario:

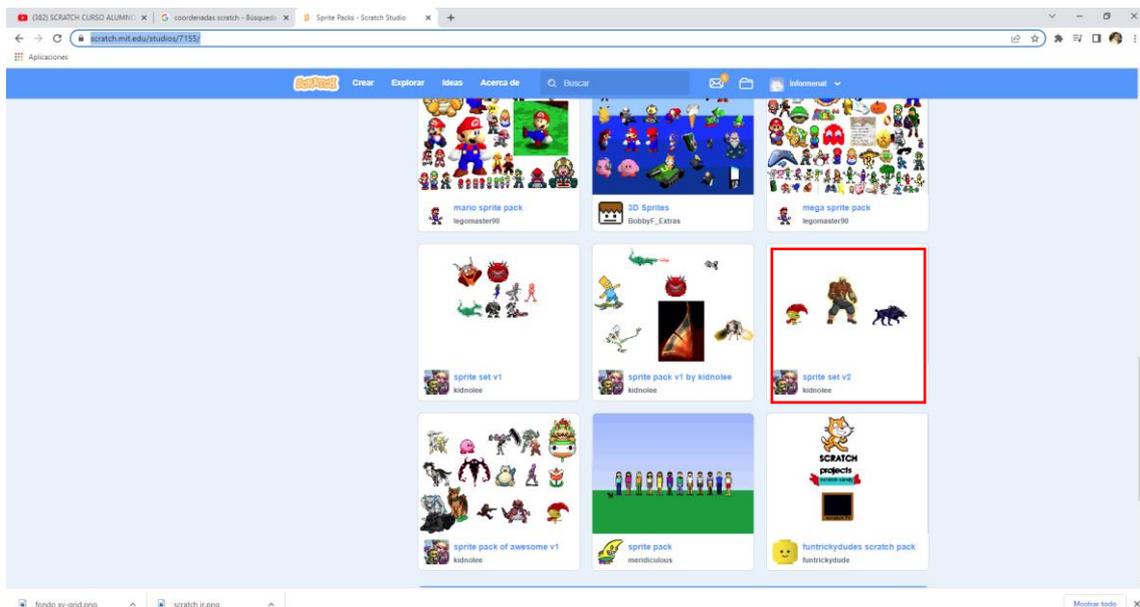


Los disfraces son las distintas formas de poder ver a un personaje, cuando programamos un objeto o personaje este funcionará indistintamente dependiendo del disfraz que tengamos seleccionado a nuestro personaje.

Con el editor de imágenes podemos modificar los disfraces.

Como importar sprites de otros proyectos.

Desde el enlace <https://scratch.mit.edu/studios/7155/> accedemos a otros proyectos que podemos utilizar.



Como ejemplo selecciono el proyecto que tengo remarcado.



Selecciono el botón Ver dentro.



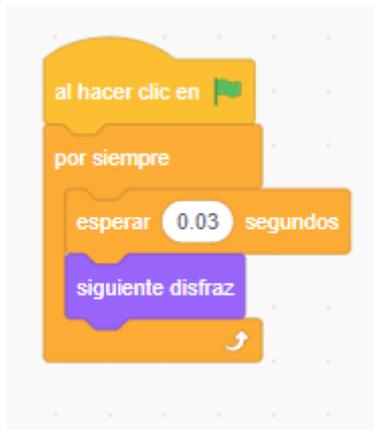
Selecciono con el botón derecho el personaje que deseo exportar. Ahora abriremos nuestro proyecto.



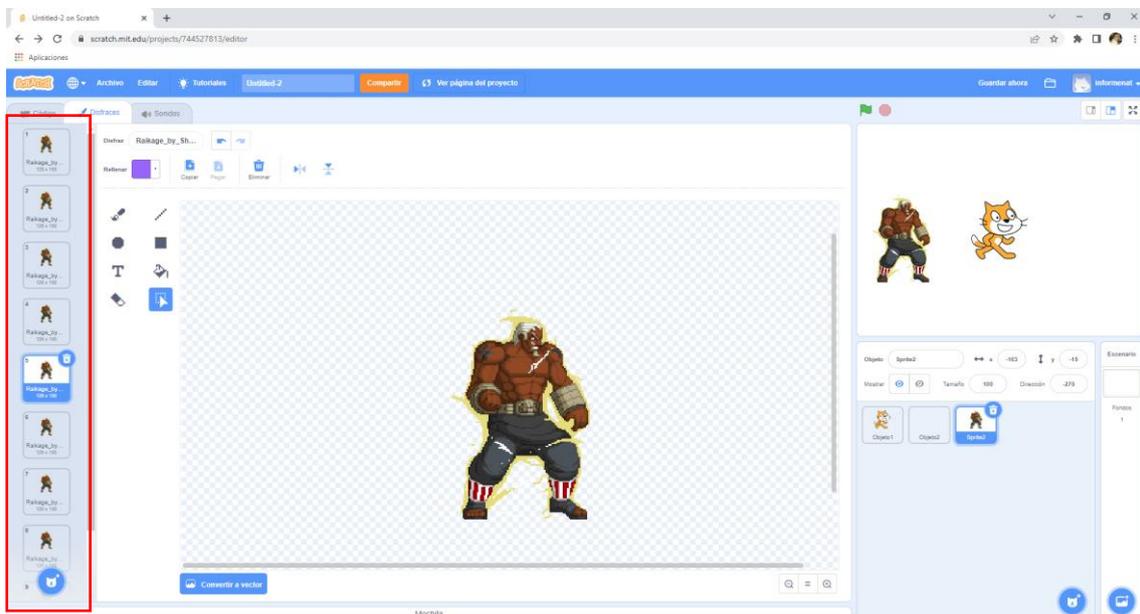
Vamos a subir el objeto que exportamos con anterioridad.



Además tenemos todo su código:



Y además todos sus disfraces.



Una forma muy útil para compartir Sprites.

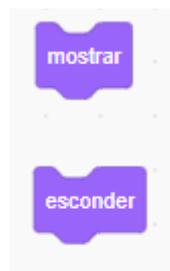
### Programar la Apariencia Objetos/Personajes

Vamos a cambiar la apariencia de nuestro objeto a través de programación, veamos las instrucciones y opciones que tenemos:

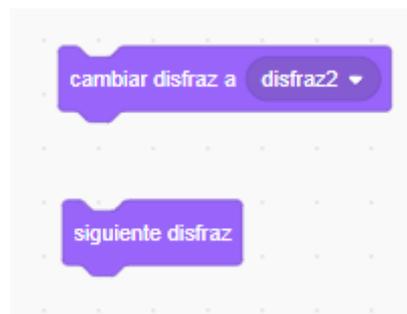
- **Decir Mensajes por Pantalla.** El personaje muestra o dice un mensaje por pantalla.



- **Mostrar u ocultar un objeto.** Pensar que todos los objetos los tenemos que crear al principio cuando creamos nuestro programa, pero quizás no todos van a aparecer en ese momento sino que aparecerán según avanza el mismo, por ejemplo distintas pantallas.



- **Programación de disfraces.** Cambiar a un disfraz en concreto o pasar el siguiente disfraz.



- **Aplicar Efectos a un personaje.** Aplicar Efectos a un personaje: cambiar el color, y otros efectos (ojo de pez, remolino, pixelar, mosaico, brillo, desvanecer). Podemos cambiar (incrementar un porcentaje a dicho efecto) o establecer que poner un valor determinado. Tenemos la instrucción quitar gráficos para dejar al personaje con su aspecto original.



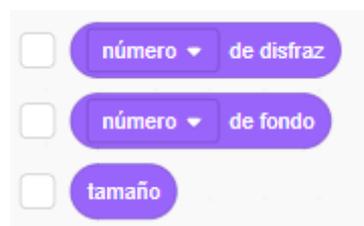
- Cambiar o Fijar el tamaño del personaje.



- Gestión de capas. Enviar al frente o mover x capas hacia atrás. Sirve para superponer objetos.



- Valor de las variables: nombre del disfraz, nombre del fondo y tamaño sobre la apariencia del personaje.



## FONDOS/ESCENARIO

- **CREAR UN NUEVO FONDO**
  - Subir un Fondo. Puede ser una imagen.
  - Fondo Sorpresa.
  - Dibujar Fondo. Nos permite dibujar nuestro propio fondo en un editor de imágenes.

- Elige Fondo. Utiliza uno de los fondos ya incorporados en la biblioteca de Scratch.
- **CONFIGURAR/EDITAR FONDO**
  - Nombre del fondo.
  - El orden es importante para pasar al siguiente fondo.
  - Edita el fondo, con el EDITOR DE IMÁGENES

### Programar la Apariencia: Fondos/Escenario

Vamos a cambiar la apariencia de nuestros fondos de Escenario a través de la programación.

- **Programación de fondos.** Cambia a un fondo en concreto o pasa al siguiente fondo.



- **Aplicar Efectos a un fondo.** Aplicar Efectos a un fondo: Cambiar el color, y otro efectos (ojo de pez, remolino, pixelar, mosaico, brillo, desvanecer). Podemos cambiar (incrementar un porcentaje dicho efecto) o establecer que pone un valor determinado. Tenemos también la instrucción quita efectos gráficos para dejar el fondo con su aspecto original.



## 5.- Estructura Básica Programación

### Eventos

Los eventos son estructuras de programación que permiten alterar el hilo normal de ejecución de un programa y pasar a ejecutar un código específico cuando se produce la condición que desencadena el evento por parte del USUARIO. Estos eventos de USUARIO en Scratch son:

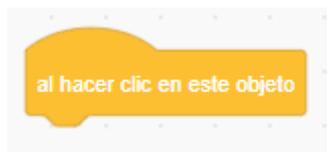
- Al hacer clic en la bandera verde.



- Al presionar una tecla.



- Al hacer clic en este objeto.



### Mensaje entre objetos

Ya sabemos que en Scratch la programación de cada objeto es propia e independiente de la del resto de los objetos. El problema surge cuando desde un objeto queremos desencadenar una determinada funcionalidad en otro objeto, esto se denomina **SINCRONIZACIÓN ENTRE OBJETOS**, para esto surgen los MENSAJES.

Los mensajes nos permiten poner en comunicación dos objetos a través del envío y recepción de un mensaje con un nombre específico, a través de las siguientes instrucciones:

- **Al recibir Mensaje.** Ejecuta el programa que tiene debajo cuando recibe un mensaje específico “enviar a todos” (Broadcast).



- **Enviar Mensaje.** Envía un mensaje a todos los Objetos y luego continua con el bloque siguiente **sin esperar a que se realicen las acciones de los Objetos activados.**



- **Enviar Mensaje y esperar.** Envía un mensaje a todos los Objetos, activándolos para que hagan algo y **espera a que todos terminen antes de continuar con el siguiente bloque.**



### Estructuras condicionales

Las instrucciones condicionales nos permiten a partir de análisis de **UNA CONDICIÓN** ejecutar cosas diferentes en función de si la condición se cumple o no. También está la instrucción que ejecutará la o las instrucciones, solo si se cumple la condición:



### Estructuras repetitivas

Las estructuras repetitivas permiten repetir un conjunto de instrucciones bajo alguna de las siguiente condiciones:

- **Repetir por siempre.**



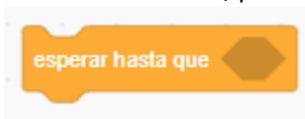
- **Repetir un numero (fijo) determinado en ocasiones.**



- **Repetir hasta que** se cumpla la condición.



- **Espera hasta que** la condición sea verdadera, para ejecutar los bloques siguientes.



## 6.- Movimientos de Personaje Usuario

### Posicionar objetos en la pantalla

#### SITUAR LOS OBJETOS/PERSONAJES EN SU POSICIÓN

En un programa tanto al principio como durante el desarrollo del mismo necesitamos situar los objetos en su posiciones correctas.

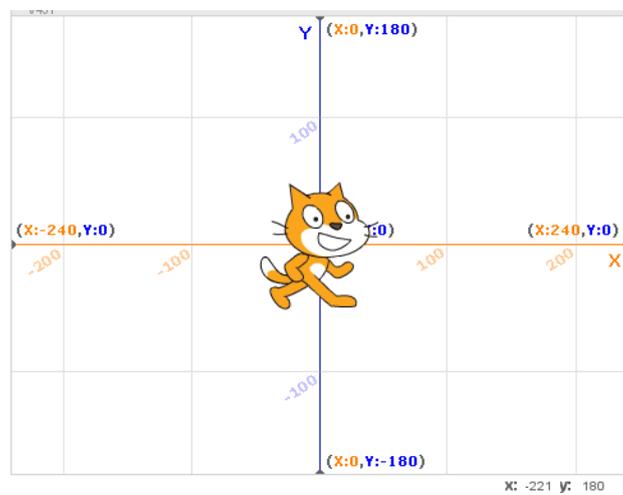
Recordar que es muy importante posicionar al principio de nuestro programa todos los objetos en su posición original, a través de programación, tanto si es una posición determinada como aleatoria. **Cuando un programa Scratch finaliza los objetos terminarán en diferentes posiciones a las iniciales** y si no volvemos a situar en su posición actual al comenzar de nuevo el programa el resultado no será el deseado.

Veamos como podemos realizar todo esto...

#### SISTEMA DE COORDENADAS EN SCRATCH

En Scratch podemos situar un objeto en cualquier lugar de la pantalla. En primer lugar tenemos que saber que la posición de los objetos viene determinada por dos valores numéricos x, y, de forma similar al sistema de coordenadas cartesiano:

- X. EJE HORIZONTAL. Valores entre -240 y 240.
- Y. EJE VERTICAL. Valores entre -180 y 180.
- La coordenada (0,0) es el centro de la pantalla.



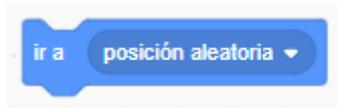
#### POSICIONAR OBJETOS EN LA PANTALLA: DETERMINADA VS ALEATORIA

Tengo 2 opciones:

- **Ir a una posición determinada**



- **Ir a una posición aleatoria.** En este caso el programa determina una posición aleatoria para el objeto, **muy útil para cuando quiero situar en videojuegos objetos en posiciones diferentes.**



## POSICIONAMIENTO CON DESPLAZAMIENTO

Si queremos posicionar los objetos en una posición en la pantalla pero que no aparezcan de repente en esta posición, si no estos se desplacen progresivamente hasta dicha posición utilizaremos las instrucciones equivalentes a las anteriores pero con de deslizar:



### Movimientos objetos controlado por el usuario

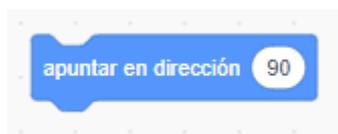
Las dos opciones más habituales es que el usuario de nuestro programa pueda mover los objetos a través del teclado o del ratón:

## MOVIMIENTO A TRAVÉS DEL TECLADO

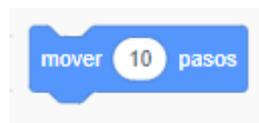
1. Partimos de la instrucción de evento al presionar la tecla:



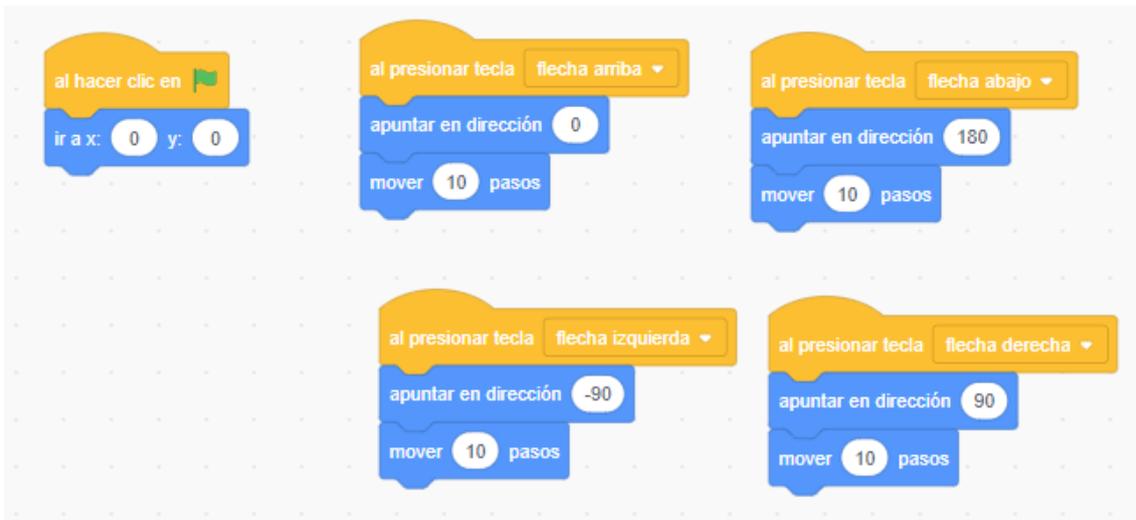
2. Ahora antes de realizar el movimiento podemos hacer que nuestro personaje apunte en la dirección en la que se va a mover, esto lo hacemos con la siguiente instrucción:



3. Finalmente realizamos el movimiento. Una vez apuntado el personaje en la dirección del movimiento simplemente hacemos un movimiento de pasos:

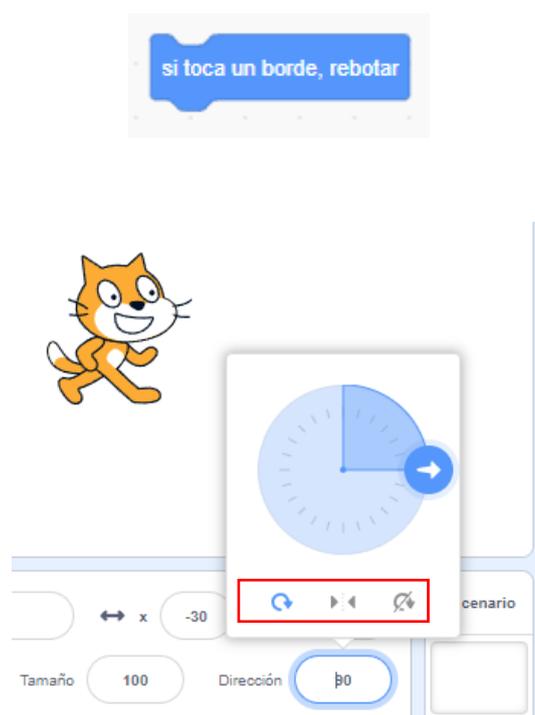


## EJEMPLO:



## REBOTES Y CONFIGURACIÓN DE LA ORIENTACIÓN DE DIRECCIÓN

En los movimientos anteriores podemos hacer que el personaje al llegar a uno de los bordes rebote, para que no continúe avanzando y además configurar la orientación de dirección para que se ajuste a nuestras necesidades.



El primero nuestro persona rota a la dirección a la que se desplaza, el segundo solo cambiar en horizontal y el tercero no realiza ninguna rotación el personaje siempre apunta a la misma dirección.

**¿Como hacer que nuestro personaje rebote?**



Ahora nuestro personaje ya no sale de los límites de la pantalla.

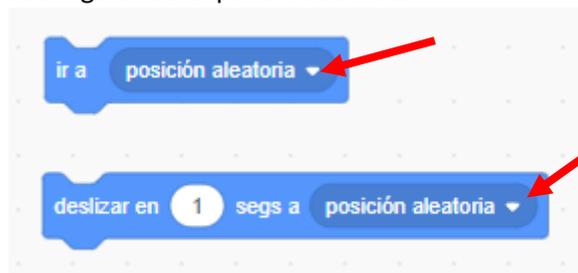
### SALTO EN VERTICAL

Para que el personaje pueda realizar un salto en vertical al pulsar una tecla el usuario, deberemos de hacerlo de una manera progresiva para que resulte más suave. Lo que hacemos, por ejemplo, es dividir el salto final, en varios saltos más pequeños, como muestra el siguiente ejemplo:

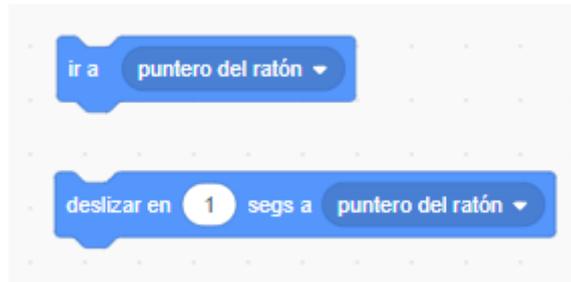


### MOVIMIENTO A TRAVÉS DEL RATÓN

El personaje de desplazará donde se encuentra el ratón lo haremos con alguna de las instrucciones siguientes configurando el puntero del ratón.



Si cambiamos esta opción a



Recordar si queremos hacer esto siempre deberemos de poner estas instrucciones dentro de un por siempre...



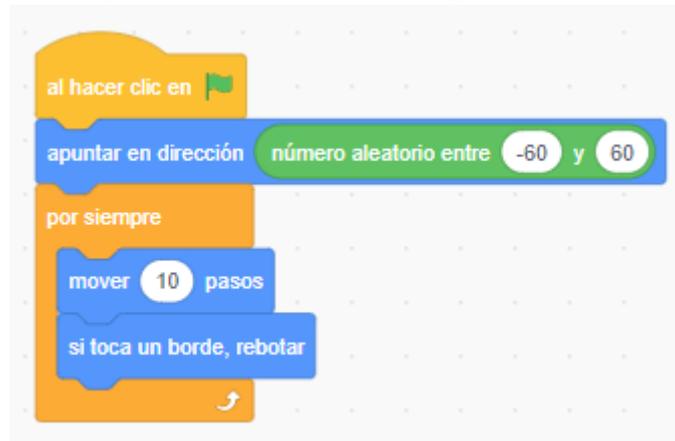
## 7.- Movimiento de Objetos solos

### Movimientos automáticos de un Objeto: Pelota

Estos movimientos incluyen que el objeto:

- Se mueve solo
- Constantemente
- Rebote si toca uno de los bordes.

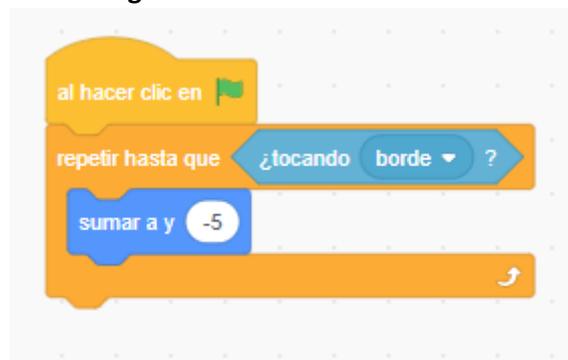
Una posible programación para este escenario sería el siguiente:



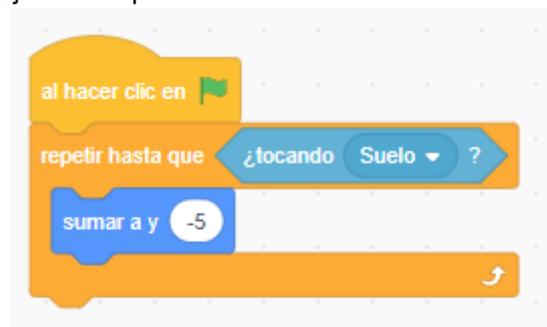
### Movimientos automáticos de un Objeto: Gravedad

La gravedad es la caída de objetos en su coordenada Y, hasta que se cumpla una determinada condición. De forma general la simulación restando constantemente un valor a la posición Y del objeto. Lo importante es determinar la condición de parada que vendrá dada dependiendo de cada caso:

- **Caída constante hasta llegar al borde inferior.**



- **Caída constante hasta tocar un objeto en el borde inferior.** En este caso puede ser que pongamos un objeto en la parte inferior.



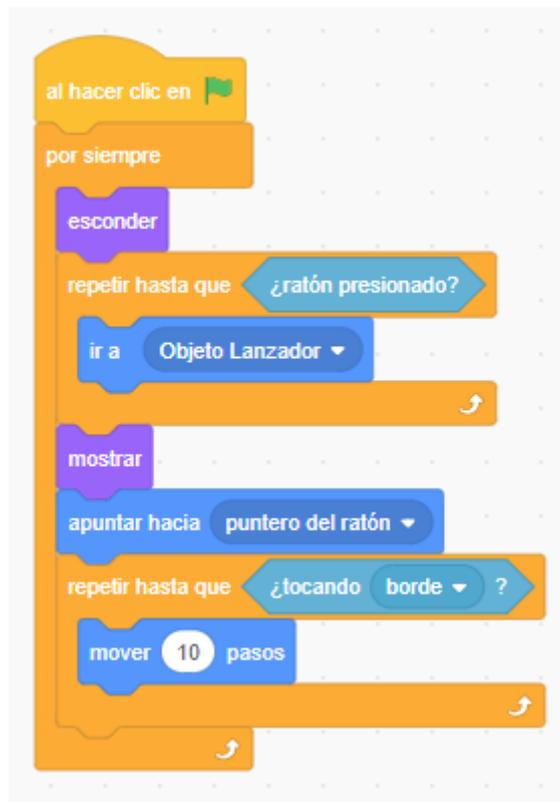
- **Caída contante hasta tocar un elemento que no es un objeto.**

En este caso podemos pintar los objetos con los que tocar y la parte de mi personaje con el color específico y hacer el movimiento mientras que ambos colores no entren en contacto.



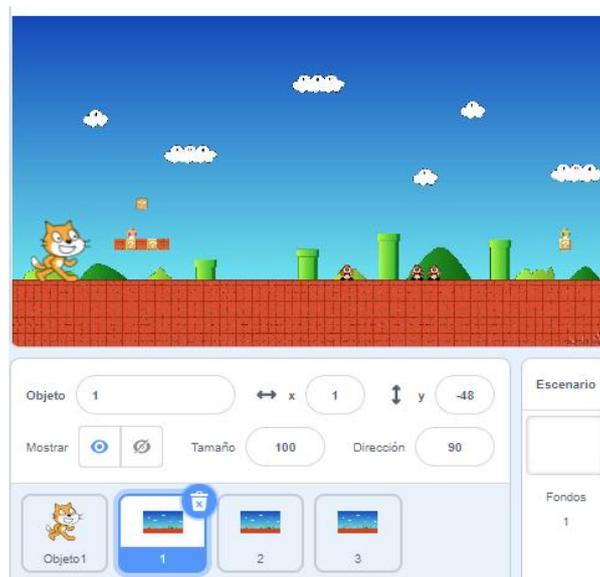
Lanzamiento de un objeto por otro objeto: Proyectoil

El programa del objeto proyectil sería el siguiente:



## 8.- Movimiento Escenario SCROLL

Vamos a importar 3 fondos como objetos y los ubicaremos en la misma posición además de un personaje que puede ser el gato.



### Scroll Horizontal de Fondo

La forma más fácil es el siguiente proceso; ahora no vamos a mover al personaje sino que movemos **los objetos que vamos a crear como fondos que van a ir uno a continuación de otros**:

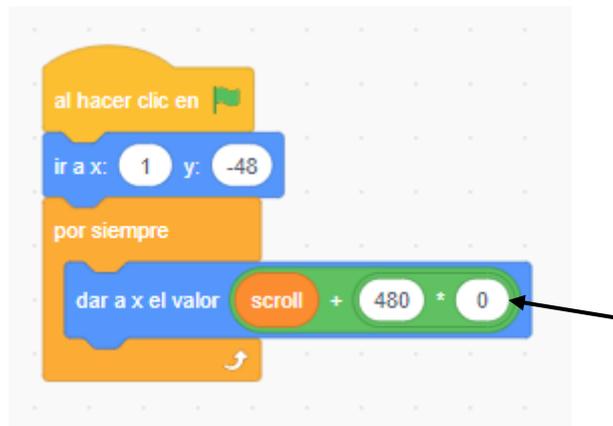
- Definimos una variable scroll, que contendrá la posición de objeto en la pantalla. Comenzamos en el código del personaje, enviándole al frente (para que aparezca por delante de los fondos), e iniciando la variable scroll a 0. Ahora también en el código del personaje hacemos que si se pulsa la flecha a la derecha cambie la variable scroll por 5, y se espera 0.1 segundos, y recíprocamente que si se pulsa la flecha a la izquierda cambien la variable scroll por -5, y si queremos añadimos una espera de 0,1 segundos.



- Ahora para cada uno de los objetos que hemos definido como fondos y por orden le asignamos el siguiente código:
  - Le asignamos la posición original a este objeto fondo.
  - Por siempre

- Fijar la x a la variable scroll + 480 \* A. Dónde A será 0 para el primer escenario, 1 para el segundo, 2 para el tercero y así sucesivamente.

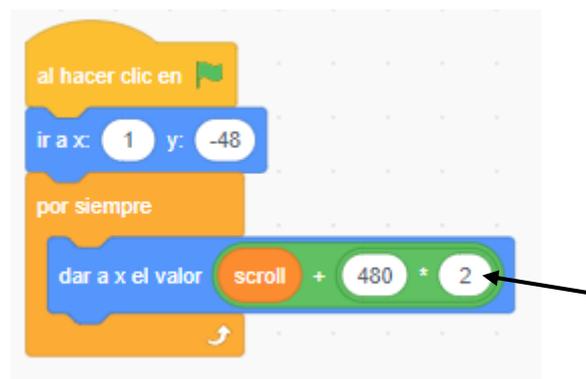
Para el primer escenario



Para el segundo escenario



Para el tercer escenario



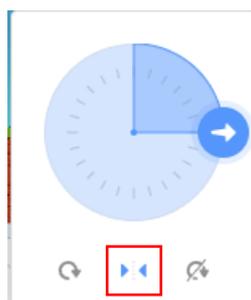
- Si en este programa se presenta un pequeño error en el “empalme de la derecha” podemos “camuflar” haciendo un marco a lo largo de toda la pantalla que lo tape.

Para controlar que los escenarios no de muevan más de lo necesario y mostrar el fondo blanco realizaremos los siguiente:



Vamos a agregar varias modificaciones, la primera si nos vamos a la izquierda que mire para la izquierda y si nos vamos a la derecha que mire a la derecha.

Realizaremos cambio de disfraz para ver el efecto de que camina.



Para que solo cambie en modo horizontal y nuestro personaje no rote seleccionaremos en propiedades de la dirección la segunda opción.

### Scroll Vertical de Fondo

Se desarrolla de forma análoga al scroll horizontal pero trabajando con la coordenada VERTICAL (Y), en vez de la HORIZONTAL (X).

Recuerda que en vertical estamos trabajando en 360 pixeles.

## 9.- Contacto Personajes COLISIONES

### Concepto de Colisión

En programación y sobre todo en programación de videojuegos se denomina colisión a la acción por la que dos elementos gráficos diferentes de nuestro programa se tocan y esto requiere una acción específica por parte de nuestro programa.



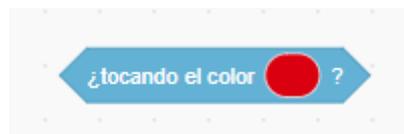
### Programar una colisión en Scratch

La programación será diferente en función del tipo de elementos gráficos que entren en contacto:

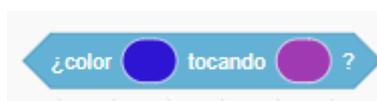
- Si los elementos que queremos evaluar si se están tocando son objetos lo mejor es utilizar la instrucción tocando e indicando el objeto.



- Si uno de los elementos no es un objeto se hace con la instrucción ¿Tocar el color \_\_\_?. Tenemos que tener cuidado que dicho color no aparezca en otra zona del programa que no sea ese.



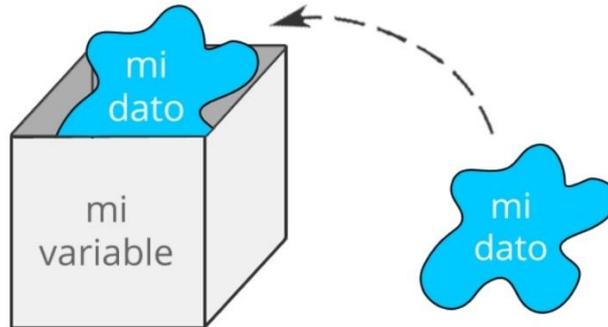
- También, y para una mayor precisión, se podría hacer identificador dos colores únicos en la aplicación que se pondrían en cada uno de estos elementos y utilizar la instrucción ¿Color \_\_\_ tocando \_\_\_?



## 10.- Guardar datos: VARIABLES

### Concepto de Variable

Una variable en un programa no es más que un espacio en la memoria del ordenador en el que se almacena un VALOR. (Metáfora de cajón de un archivador).



Podemos asignar en cualquier momento el VALOR correspondiente en dicha variable (Ejemplo: PUNTOS=10, VIDAS=2)

Para referirse al valor almacenado es necesario asignar un NOMBRE a dicha variable.

### TIPOS DE VARIABLES

- Locales. Afecta solo a el objeto que la define, el resto de objetos no tiene acceso a la misma.
- Globales. Tanto el objeto que la define, como todo el resto de objetos tienen acceso a la misma.

### Variable en Scratch

Mediante el bloque variables, puedo utilizar la funcionalidad de variables en Scratch.

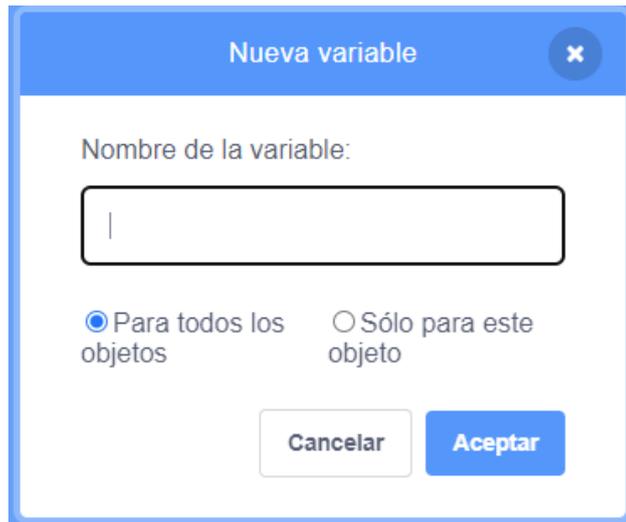
#### Al crearla definimos el tipo de variable:

- Para todos los objetos. Variable Global.
- Solo para este objeto. Variable Local.
- Opción de Variables en la Nube. El valor de la variable se guarda en la nube aunque cerremos nuestro programa.

Solo permite almacenar números y está reservada para usuarios **Scratcher**.

Cuando se une un nuevo usuario, se le asigna el estado "Nuevo Scratcher". Esto plantea restricciones, que incluyen no poder usar Cloud Data y esperar 2 minutos entre publicaciones en los foros de discusión. Más se puede ver aquí. Para obtener el estado "Scratcher", un Nuevo Scratcher debe:

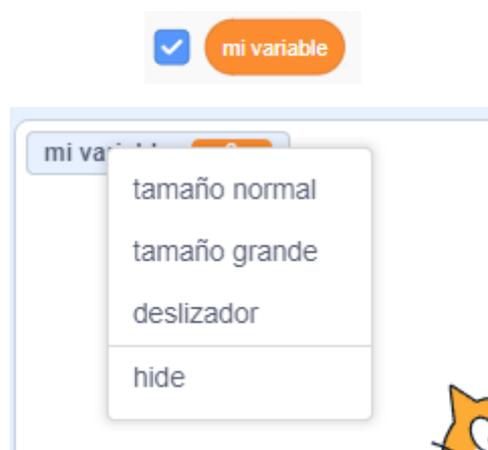
- Tener una dirección de correo electrónico verificada.
- Tener 2 proyectos compartidos.
- Ha estado en Scratch durante un cierto periodo de tiempo (al menos 2 semanas, pero generalmente 1 mes)
- Mantener activo el sitio web. Esta es una gran parte de convertirse en un Scratcher. Compartir proyectos regularmente puede ser una forma.
- Tener comentarios y/o publicaciones en el foro. (Esto no es necesario, pero ayuda).



- Podemos asignar valores y leer los valores de dichas variables.

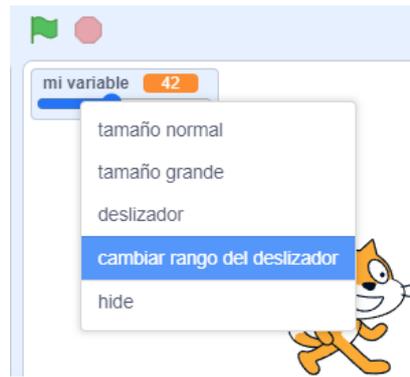


- Si activamos el checkbox que aparece al lado del nombre de la variable que hemos definido o la instrucción correspondiente de mostrar variable, podemos conseguir que dicha variable aparezca o no en la vista del programa. Además pinchando sobre esa vista de la variable podemos cambiar el aspecto con el que se muestra en la pantalla dicha variable.



Con la opción deslizador permitimos que el usuario pueda cambiar su valor.





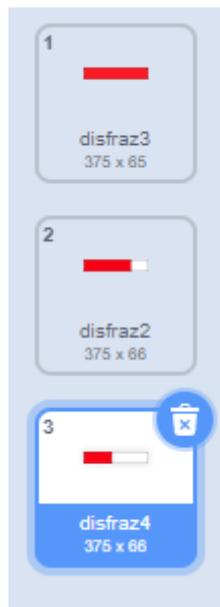
Podemos cambiar el rango.

### Barras de puntuación y dibujos con vidas

Para hacer cualquiera de estas dos funcionalidades típicas de videojuegos utilizamos la estrategia de los disfraces.

#### BARRA DE NIVEL

Definimos un objeto con el dibujo de barra y deberemos de incorporarle varios disfraces. En cada uno de esos disfraces se muestra la barra disminuyendo progresivamente su nivel. Cada vez en el juego que tengamos que reducir el nivel de la barra pasaremos al siguiente disfraz.



#### DIBUJOS CON VIDAS

En este caso debemos de poner tantos objetos con la misma imagen, como vidas tengamos. Al principio de todo estos objetos son visibles, cada vez que perdemos una vida, en función del número de vidas ocultaremos la vida correspondiente.

1  
Heart Red  
418 x 90

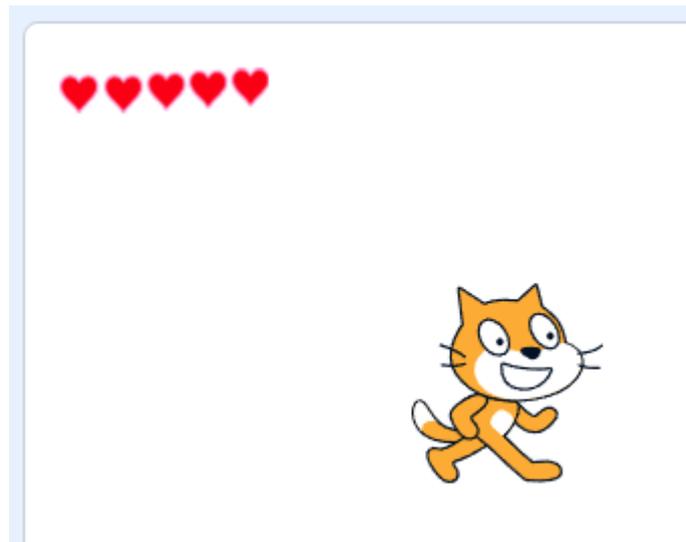
2  
Heart Red2  
334 x 86

3  
Heart Red3  
251 x 82

4  
Heart Red4  
166 x 77

5  
Heart Red5  
78 x 77

6  
Heart Red6  
0 x 0



## 11.- Sonido y Música

### Sonido

Puedo utilizar sonidos asociados a los objetos con los que voy a trabajar dentro del bloque de SONIDO.

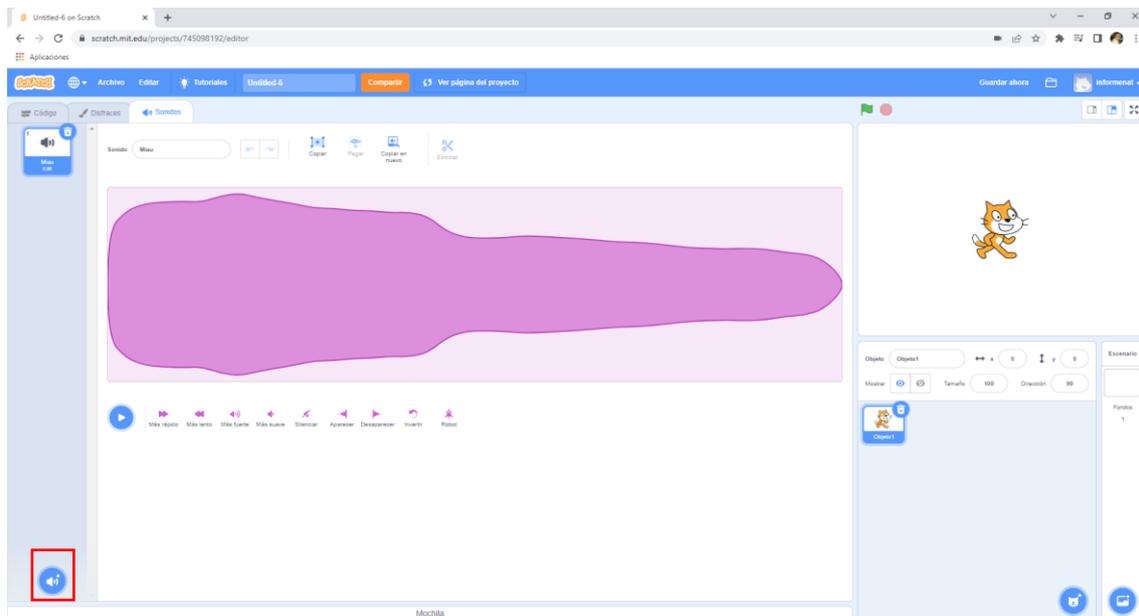
#### ASIGNACIÓN DE SONIDOS A UN OBJETO.

Debemos asignar los sonidos que queramos al objeto correspondiente, estos sonidos pueden tener el siguiente origen:

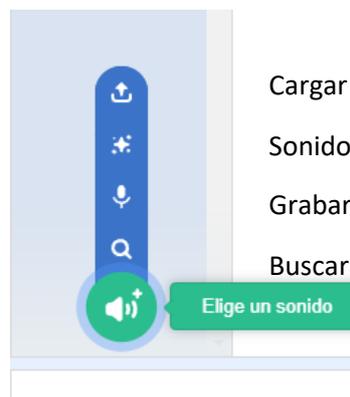
- Cargar un sonido de una archivo.
- Aleatorio (sorpresa).
- Grabar directamente del micrófono del ordenador, me puedo grabar mi voz.
- Elegir de la biblioteca de Scratch.

#### EDITOR DE SONIDOS

A través de este editor podremos realizar operaciones básicas de audio sobre los diferentes sonidos con los que estamos trabajando:



En cada objeto tenemos tres pestañas Código, Disfraces y Sonido, desde esta pestaña podemos asociar todos los sonidos que necesitamos para un elemento o personaje.



Cargar un sonido.

Sonido sorpresa.

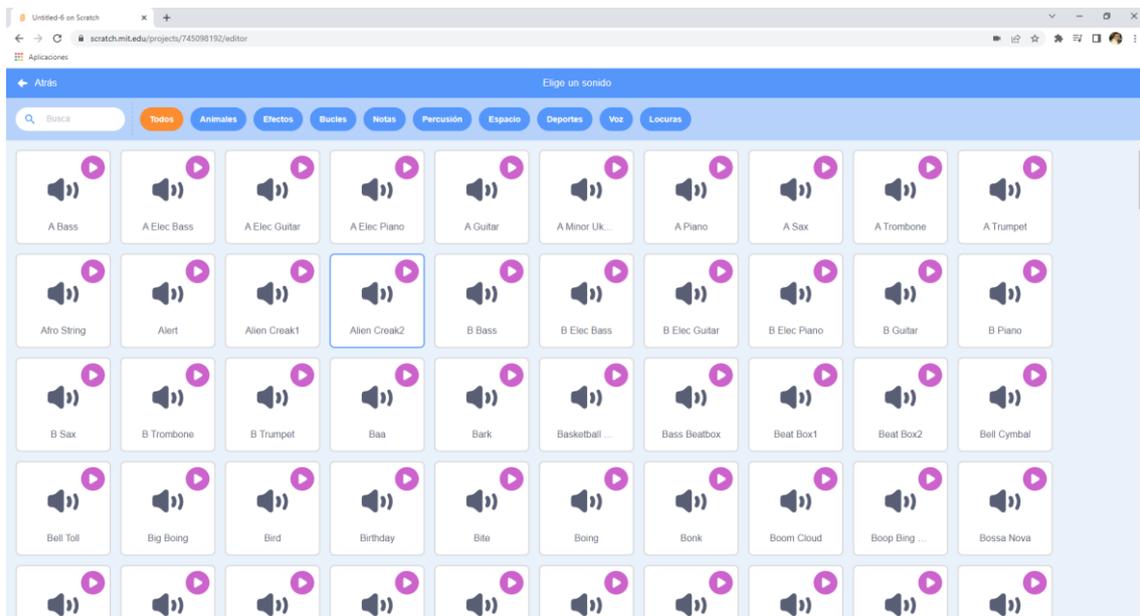
Grabar desde el micrófono de mi Pc.

Buscar en la galería de sonidos.

Si accedemos a grabar desde el micrófono de mi Pc.



Si seleccionamos buscar en galería de sonidos:



Podemos agregar una gran cantidad de sonidos que están agrupadas por tipos de sonido.

Si te sitúas encima de cada Play sin presionar podremos reproducir el sonido.

Podemos agregarla a nuestro personaje.

### INSTRUCCIONES CON SONIDOS

- **Iniciar y tocar sonidos y parar sonidos.** La diferencia entre iniciar y tocar, es que la segunda no continua con la ejecución del programa hasta que se ha terminado de reproducir todo el sonido. Se realizan con las siguientes instrucciones.



- **Aplicar efectos sobre los sonidos.** ALTURA (niveles de graves y agudos) y el BALANCE IZQUIERDA/DERECHA (lugar por donde escuchamos los sonidos).



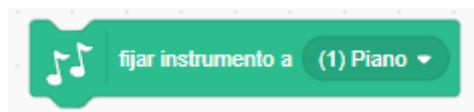
- Además en todo momento puedo modificar y consultar la **variable global volumen**.



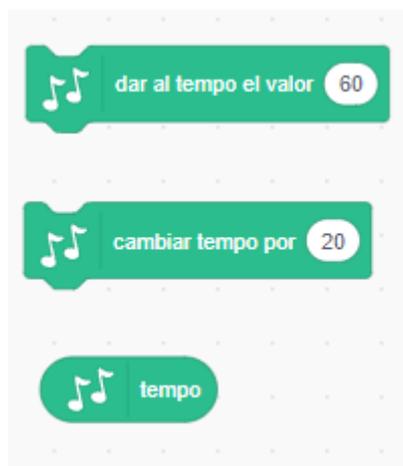
## Música

Scratch permite trabajar con música a través de la **EXTENSIÓN MÚSICA**, que usará el apartado MIDI de nuestro PC. Esto nos permite tocar notas musicales de diferentes instrumentos.

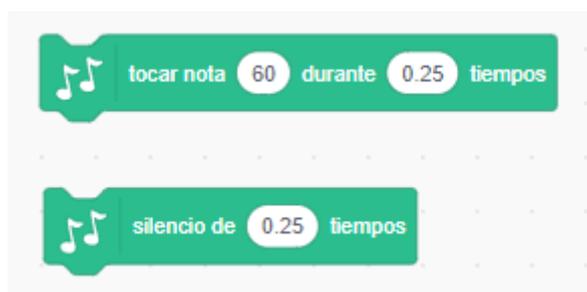
- Podemos fijar el **instrumento** con el que queremos tocar.



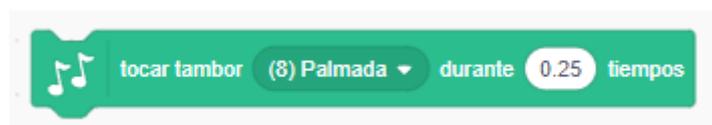
- Podemos fijar o cambiar (incrementar) el **tempo**. El Tempo es la velocidad (bpm = beats por minuto) a la cual se tocan las notas y tambores en Scratch. Cuanto mayor sea el tempo, más rápido sonará las notas y tambores.



- Con tocar **nota**, le indicamos la nota que queremos que toque y la duración en tiempo de la misma (esto equivale a la duración de las notas blanca, negra, pero evidentemente este depende del tempo que tengamos fijado que son las pulsaciones por minuto). Pinchando sobre la nota sale un teclado dónde podemos indicar la nota correspondiente (tenemos dos escalas enteras, en el piano que aparecen para seleccionar las notas pero podemos seguir la poniendo números a mano anteriores al 48 y posteriores al 72 para poder ir ampliando las escalas).



- La **percusión** en MIDI va por otro lado, con una lista diferente de instrumentos de percusión con la instrucción tambor.



- Se puede hacer **acordes** o coros, colocando dos líneas de ejecución (al presionar bandera verde) y se irán tocando las dos secuencias de notas a la par.

**Obs. Además en todo momento puedo consultar las variables globales volumen y tempo para ver sus valores.**

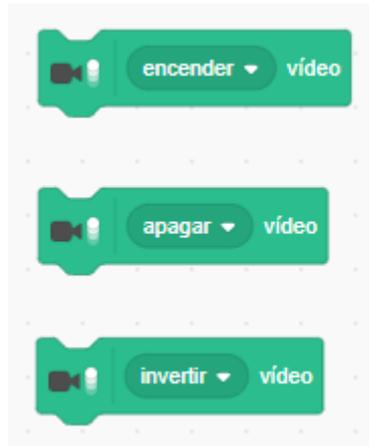


## 12.- Cámara y Micrófono

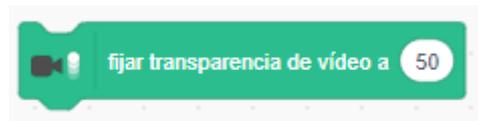
### Cámara

Puedo utilizar la funcionalidad de la WebCam dentro de un programa de Scratch. Para ello es necesario activar la EXTENSIÓN sensor de vídeo. Podemos gestionar el control de la Webcam con los siguientes comandos:

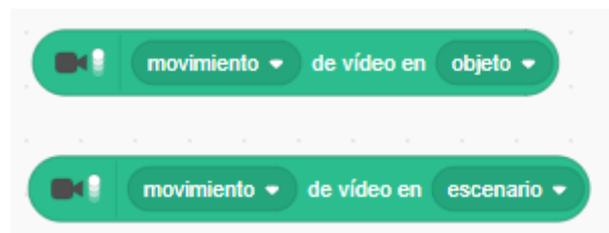
- **Encender, apagar e invertir cámara** (invierte la posición horizontal de la cámara).



- Controlar el **nivel de transparencia** con la que se va a ver nuestra imagen en el fondo.



- **Movimiento del vídeo en:** este objeto o sobre el escenario, que nos permite obtener un valor de 0-100 con el nivel de movimiento sobre este objeto.



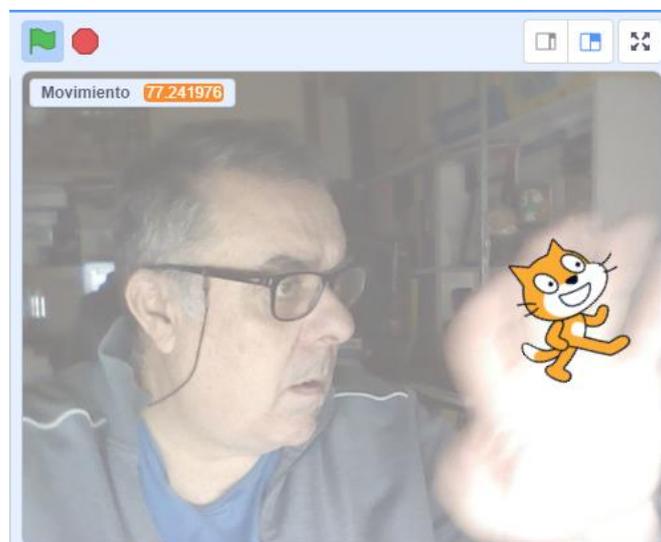
- **Dirección del vídeo:** (cambiando la opción de la instrucción anterior). Me permite apuntar en la dirección en la que estoy apuntando a través de la WebCam. Esto se haría: apuntar en dirección, dirección de vídeo en este objeto.



Ejemplo:



Nuestro objeto va bajando pero cuando nos movemos por detrás de la imagen este objeto deja de bajar.

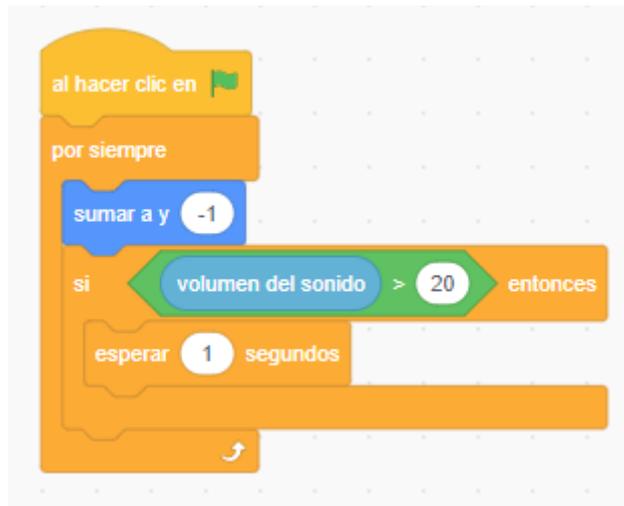


### Micrófono

Podemos utilizar el micrófono en la ejecución de un programa de Scratch para **identificar el nivel de sonido en el mismo** y actuar en consecuencia. Se realiza a través de la siguiente instrucción de bloque de sensores:



Ejemplo:



Nuestro personaje va cayendo en el momento que detecte un sonido superior a 20 el personaje dejará de caer hasta que el sonido vuelva a ser inferior a 20.

## 13.- Crear nuevos Bloques: FUNCIONES

### Concepto de Función

Las Funciones, también llamadas Procedimientos, son un **conjunto de instrucciones empaquetadas bajo el nombre de un nuevo bloque**, para que pueda ser ejecutada en diversos momentos sin necesidad de escribir todo este conjunto de instrucciones.



El objetivo de las funciones en programación suele ser el siguiente:

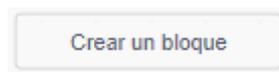
- Agrupar instrucciones que van a ser ejecutadas en diversos puntos del programa para **evitar duplicar todas las instrucciones** y así hacer el programa más reducido y comprensible.
- Agrupar instrucciones que cumplen una determinada funcionalidad, aunque no vayan a ser utilizadas en varias ocasiones, con objeto de **hacer el programa más modular y comprensible** para el programador.

### Programar Funciones en Scratch

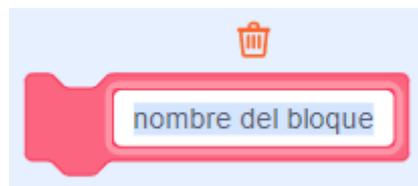
La programación de funciones se realiza a través del bloque “**Mis bloques**”.

#### 1.- CREACIÓN DE UNA NUEVA FUNCIÓN

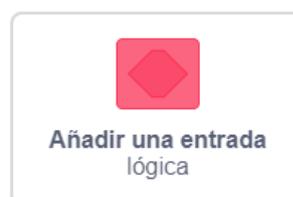
- **Seleccionaremos crear un bloque.**



- Asignamos un **nombre al nuevo bloque.**

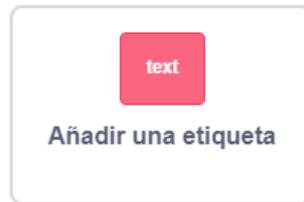


- **Añadir una Entrada.** Seleccionamos, si deseamos, los **argumentos que vaya a tener la función**. Los argumentos son valores que recibe una función y que va a poder utilizar internamente para hacer los cálculos que precise.



- Añadir una Etiqueta. Esto nos permite completar el nombre de la función para poder complementarlo con más texto, normalmente después de argumentos.

Ejemplo: SALTA Argumento1 PASOS



- **Ejecutar al instante.** Cuando crea un bloque personalizado "Ejecutar sin actuación de pantalla", reproducirá todo el script sin cambiar la pantalla hasta que finalice.

Ejecutar al instante

## 2.- PROGRAMACIÓN DE LA NUEVA FUNCIÓN

- Ahora debemos **programar el código de nuestra función** (nuevo bloque). Para ello situaremos las instrucciones correspondientes debajo del nuevo bloque que aparecerá en "Mis bloques" con el nombre de definir el nombre y argumentos de mi función.
- Para utilizar los argumentos dentro de mi función, simplemente los pincho (del encabezado) y los arrastro hasta el lugar donde los tenga que utilizar. Estos argumentos no pueden ser utilizados fuera de la función.



## 3.- USO Y ALCANCE DE UNA FUNCIÓN EN SCRATCH

Para usar el nuevo bloque, simplemente debemos de arrastrar esta nueva instrucción desde "Mis bloques" al lugar del programa donde quiera que se ejecute.

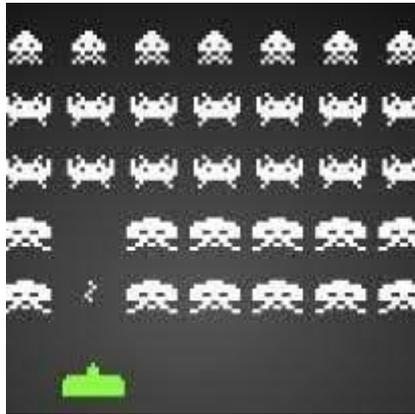
Las Funciones, nuevo bloque, que realicemos en un objeto de Scratch solo puede ser accesible por dicho objeto.



## 14.- Copias de un Objeto: CLONES

### Concepto de Clones

Una de las limitaciones de Scratch 1.4 es que era necesario crear un personaje por cada entidad. Es decir, que si quería tener 10 marcianos en un videojuego, aunque fueran todos iguales y todos tuvieran el mismo comportamiento porque ejecutar el mismo código, necesitabas programar 1 personaje y luego hacer 9 copias.



Este esquema tenía varias limitaciones, ya que hay un número máximo de personajes y además el mantenimiento y depuración del código se complica, ya que si quieres realizar cualquier cambio tiene que hacerlo en todos los personajes.

### Programar Clones en Scratch

Un **Clon** es un objeto que es idéntico a el mismo autorreplicado. La programación de funciones se realiza a través del **bloque Control**.

2. **“Al comenzar como clon”**. Aquí se programa el código que queremos que haga el clon cuando se crea.



3. **“Crear clon de”** . Crea un clon de si mismo o de cualquier OBJETO.



4. **“Eliminar este clon”**. Elimina y borra el clon.



5. Para identificar cada clon, creamos una variable para llevar la contabilidad total de clones y una local al objeto para identificar a este clon especialmente frente a los demás.

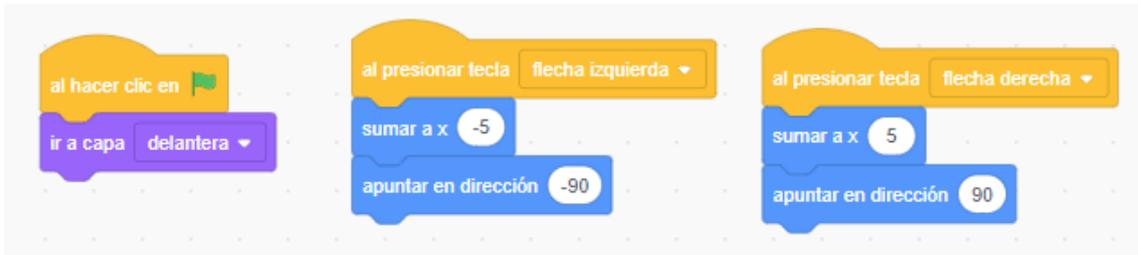
### Ejemplo:

Vamos a realizar un ejemplo donde nuestro gato va a lanzar flechas.

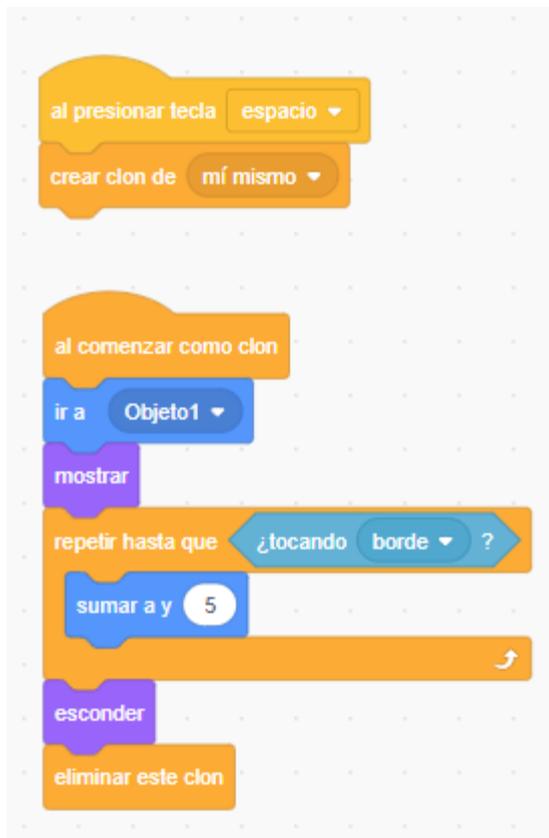
Vamos a agregar los siguientes objetos:



En el gato vamos a poner el siguiente bloque para poderlo mover con las flechas de derecha e izquierda.



Ahora este será el código de la flecha, es donde vamos a utilizar los clones.



Al empezar el programa ocultamos el objeto.

Al presionar la barra espaciadora creamos un clon de la flecha.

Al comenzar como clon:  
La flecha se tiene que posicionar a donde está el gato.  
Se tiene que mostrar.  
Vamos a realizar un bucle hasta que la flecha toque el borde.  
Sumar a y 5, esto hará que se desplace hacia arriba.

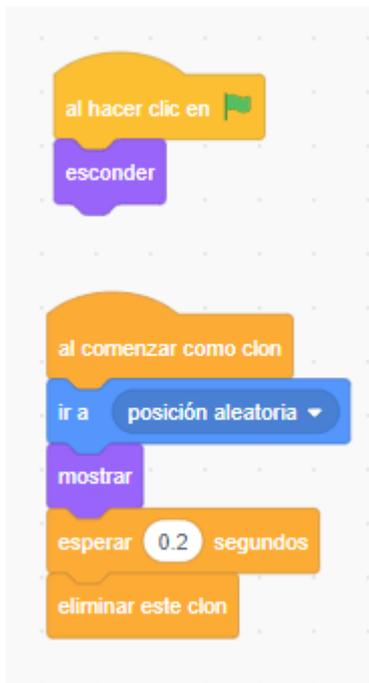
Salido del bucle esto quiere decir que ya ha tocado el borde superior, esta se tiene que esconder y para que no se acumulen los clones lo eliminamos.

Vamos a ver otro ejemplo donde el clon lo creamos desde otro objeto.

Seleccionamos el escenario y en el agregamos el siguiente bloque:



Ahora los bloques que tienen que ir en el Objeto1.



Al empezar ocultamos el objeto.

Al comenzar como clon:

Se va a una posición aleatoria.

Se muestra.

Espera 0,2 segundos.

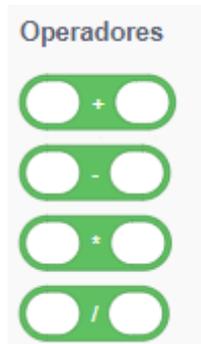
Eliminamos el clon.

## 15.- Matemáticas y Texto

### Operaciones Matemáticas en Scratch

Dentro del bloque operadores puedo encontrar los siguientes operadores matemáticos, relacionado con números:

#### 6. Suma, Resta, Multiplicación y División.



7. **Números aleatorios.** Muy útil para los juegos. Genera un número aleatorio entre los valores mínimo y máximo indicados.



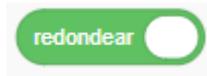
8. **Otras opciones Matemáticas.** La última instrucción (dónde pone valor absoluto de.. permite hacer muchas operaciones matemáticas más avanzadas): abs, piso, techo, raíz cuadrada, Trigonometría (sen, cos, tg ...) potencias... y logaritmos.
9. Piso y techo es redondear un número por arriba o por abajo.



10. **Hacer la operación módulo** (resto de la división de un número entre otro)



## 11. Redondear.



### Operaciones con Texto en Scratch

A los textos (cadenas de texto), también se le pueden aplicar operador, para realizar operaciones y cambios sobre las mismas. Estas operaciones (operadores) los podemos encontrar dentro del **bloque operadores**.

- Unir (concatenar) dos cadenas de caracteres.



- Extraer una letra de una cadena de caracteres.



- Calcular la longitud de una cadena de caracteres.

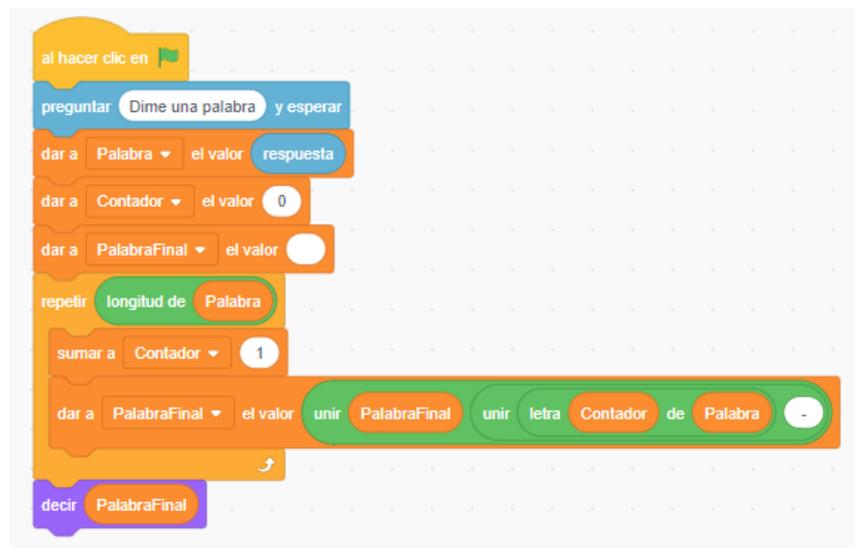
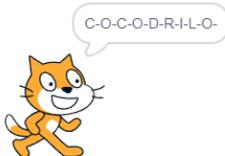


- Comprobar si una cadena de caracteres se encuentra dentro de otra.



### Ejercicio práctico:

Vamos a realizar una practica donde partiendo de una pregunta que te formula el programa, nuestro gato nos tiene que decir la misma palabra pero separada por guiones:

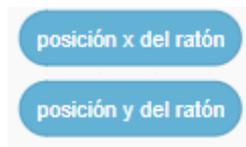


## 16.- Otros sensores

### Otras instrucciones con sensores

Dentro del **bloque de sensores** puedo encontrar algunas otras instrucciones relacionadas con esta funcionalidad que no hemos visto en ningún apartado anterior, esto son:

- **Posición del Ratón.** Indica las coordenadas x, y de la posición actual del ratón.



Con este código podremos ver en pantalla la posición que tiene nuestro ratón.



- **Permitir el Arrastre o no de objetos.** Podemos indicar si permitimos en tiempo de ejecución que se pueda arrastrar un usuario por parte del usuario.



Lo tienes que probar con el control de pantalla completa.

- **Cronómetro.** Podemos añadir un cronómetro a nuestro programa a través de la variable cronómetro. Lo podremos reiniciar en cualquier momento.



- **Información de otro objeto que no es el tuyo o del escenario.** Podemos obtener información de cualquier objeto que no sea objeto activo de programación o del escenario a través de la siguiente instrucción.



- **Año Actual.** Obtener el año actual.



Además de



Mes, día, día de la semana, hora, minuto y segundo.

- **Distancia puntero ratón.** Calcula la distancia del puntero con el objeto activo.



## 17.- Text a Voz y Traducciones

### Texto a Voz

Mediante la extensión Texto a Voz podemos reproducir textos a través del audio del ordenador, con las siguientes instrucciones:



Para este capítulo hemos creado una variable de tipo lista llamada Semana que contiene los días de la semana.

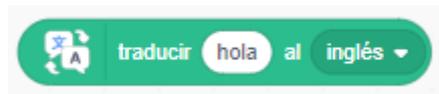
Semana	
1	Lunes
2	Martes
3	Miércoles
4	Jueves
5	Viernes
6	Sábado
7	Domingo
+ longitud 7 =	

Ahora vamos a hacer que nos digas los días de la semana en voz.



## Traducciones

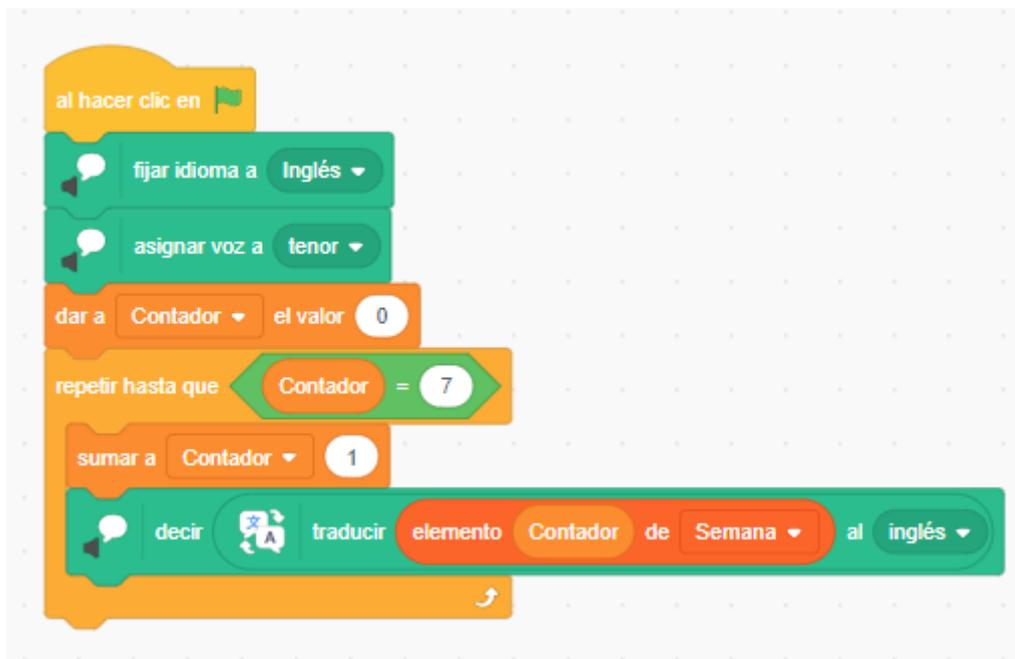
Mediante la extensión Traducción podemos mediante una sencilla instrucción traducir una expresión al idioma que queremos.



También podemos utilizar la variable idioma, para saber el idioma que tenemos configurado en Scratch.



Como ejemplo y aprovechando el código del ejemplo anterior queremos que nos diga los días de la semana en inglés.



## 18.- Dibujando con Scratch: LÁPIZ

### Concepto de Lápiz en Scratch

El Bloque de Lápiz tiene una serie de instrucciones las cuales permiten realizar dibujos en el Escenario, al usar el Objeto en el Escenario como lápiz, y por movimiento del Objeto se realiza el trazo de una línea que va del punto inicial al punto final de movimiento del Objeto. Creándose una secuencia de líneas las cuales forman la figura que uno elija.

El proceso es simple cuando se quiere asociar el Lápiz, se usa la instrucción bajar Lápiz y a partir de ahí por cada movimiento del Objeto se dibuja una línea, se deja de dibujar en el Escenario usando la instrucción subir Lápiz. Se cuenta además con una serie de instrucciones que permiten fijar características de las líneas que se trazan tales como color, tamaño e intensidad.



### Programar con el Lápiz en Scratch

Para poder trabajar con el lápiz tengo que acceder a las extensiones y seleccionar la extensión Lápiz. A partir de este momento aparece un bloque nuevo de instrucciones con las que podré realizar las siguientes operaciones:

- **Bajar/Subir el Lápiz.** Si bajamos el lápiz a partir de ese momento cualquier movimiento que realiza mi personaje va dejando un rastro con el lápiz, el dibujo parte de la posición inicial de mi personaje. Si subimos el lápiz y movemos el personaje, este dejará de dejar rastro.



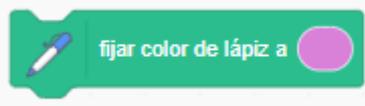
- **Borrar.** Borra la pantalla completa referente a lo que hayamos dibujado.



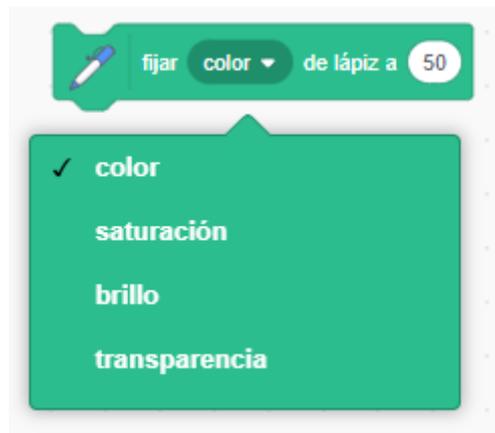
- **Sellar.** Nos deja estampada, la figura del personaje en esa posición. Con las instrucción borrar anterior se borra todo.



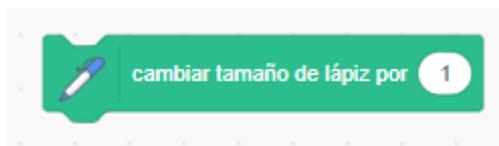
- **Fijar color del lápiz por ... un color.**



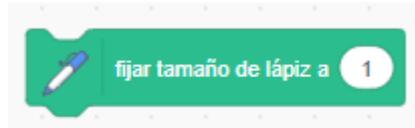
- **Fijar color/saturación/brillo/transparencia del lápiz por ...un valor numérico que corresponde a dicho color.** Color\_lápiz=0 va al extremo rojo del arcoíris, Color\_lápiz=100 va al extremo azul del arcoíris. Varía de 0 a 200 para ir alrededor del círculo cromático.



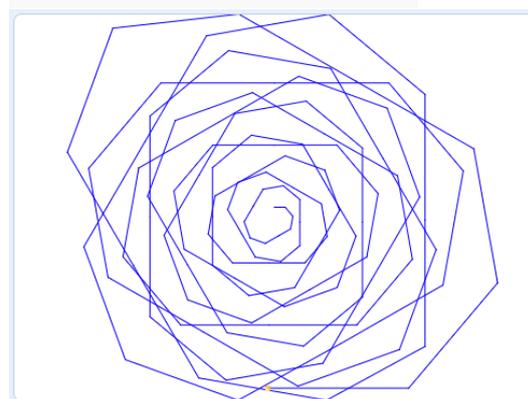
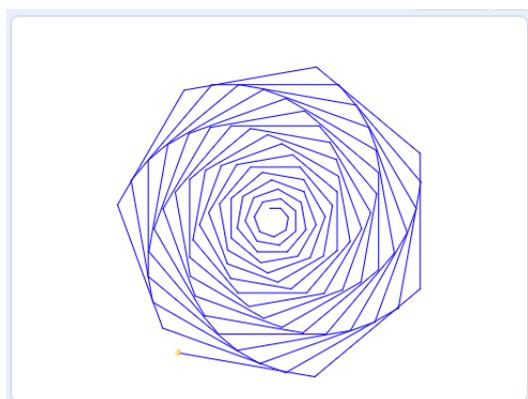
- **Cambiar tamaño de lápiz por.** Vamos a incrementar el tamaño por el valor que especifiquemos.

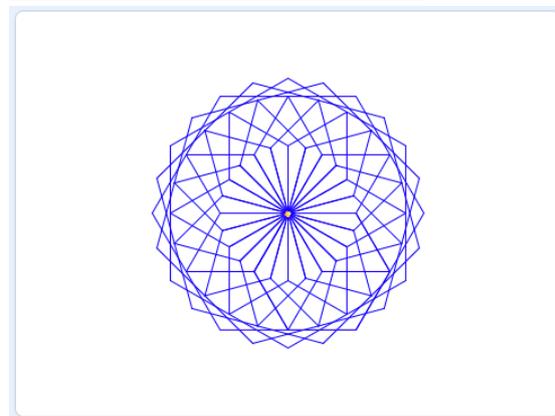
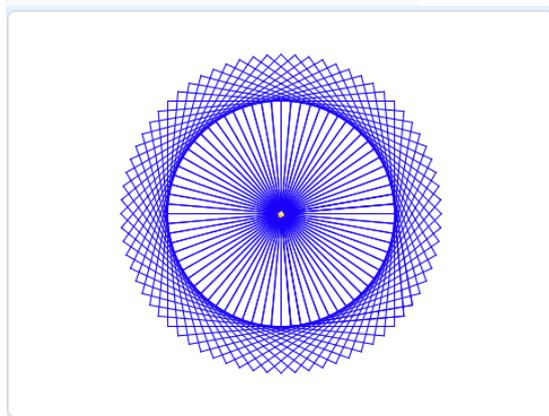


- **Fijar tamaño de lápiz a.** fija el tamaño del lápiz al valor que especifiquemos.



Vamos a realizar los siguientes dibujos:





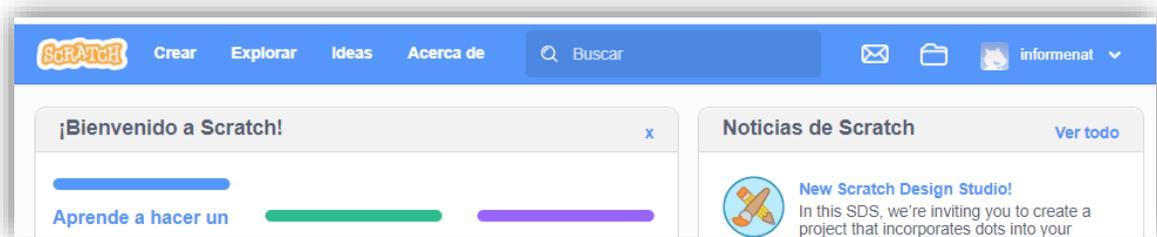
Ahora está tu imaginación para realizar otros dibujos.

## 19.- Explorar, Buscar y Compartir Proyectos

### Explorar Proyectos

Uno de los grandes éxitos de Scratch está en la importante comunidad de personas que está utilizando esta plataforma, pero más allá de su uso, está que muchas de estas personas comparten sus proyectos de tal forma que todos podemos aprender y enriquecernos de ello.

Para poder explorar estos proyectos creados por otras personas tenemos que acceder al menú Explorar de la página principal.



Ahora podremos hacer una búsqueda por proyectos o estudios:



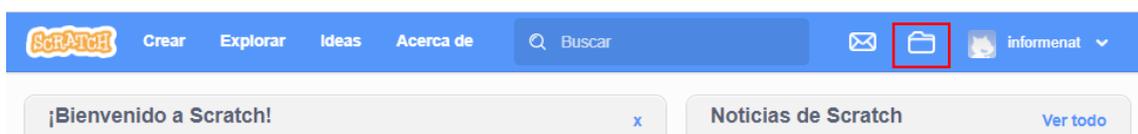
- **Proyectos.** Podremos buscar por categorías, populares, tendencias o recientes.
- **Estudios.** Los estudios son agrupaciones de proyectos que comparten una temática o tienen elemento en común.

### ACCEDER A UN PROYECTO

Una vez seleccionado el proyecto que nos interese accederemos a la ficha del mismo, donde veremos la información detallada del proyecto. Ahora tenemos dos opciones:

- **Ver dentro.** Puedo acceder al código del programa para ver como está programado y aprender del mismo.

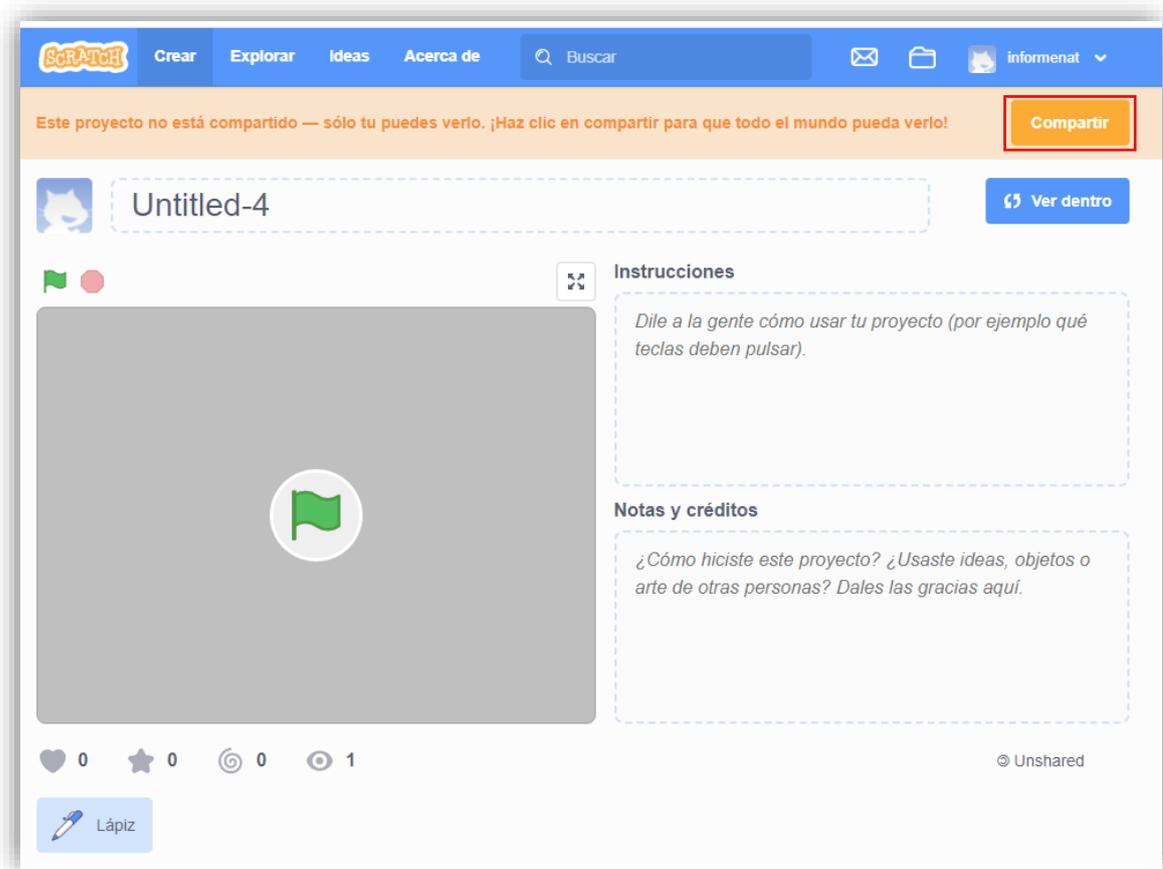
### Compartir Proyectos



Accedemos a nuestros proyectos.



Seleccionamos el proyecto que queremos compartir.



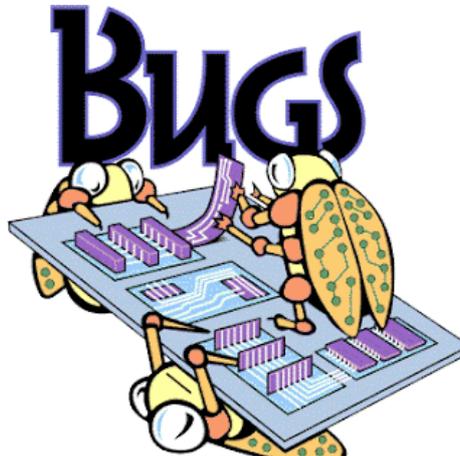
Tienes que introducir el título, comentar las instrucciones y seguidamente las Notas y créditos, seguido del botón compartir.

## 20.- Depuración de Programas: ERRORES

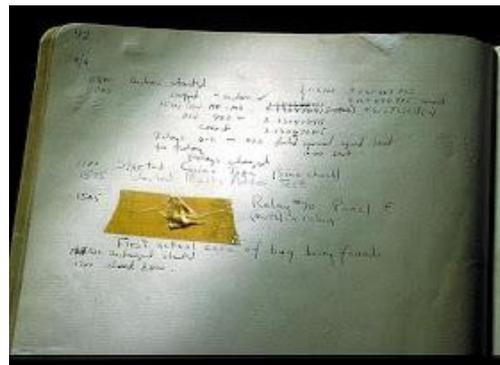
### Errores en los programas

Todo programador tiene que hacer frente a programas que parecen totalmente correctos pero que no se comportan como se espera de ellos.

Para encontrar los errores, o **bugs**, es necesario depurar los programas (**debugging**).



### ORIGEN DEL TERMINO BUG



### Errores típicos en Scratch

- **EJECUCIÓN DEMASIADO RÁPIDA.** La ejecución de instrucciones en Scratch se hace muy rápida y en muchas ocasiones de forma visible para el ser humano por lo que a veces es necesario poner instrucciones de espera.

esperar 1 segundos

- **PROGRAMAS QUE FUNCIONAN BIEN SOLO LA PRIMEVA VEZ.** Este es un problema típico por el que los programas a partir de la segunda vez se ejecutan no funcionan correctamente, el problema radica en que debemos inicializar siempre los objetos con los valores iniciales: tamaños, posiciones, valores de variables...
- **INSTRUCCIONES PROGRAMADAS QUE NO SE EJECUTAN "SIEMPRE".** A veces programamos instrucciones y vemos que no se ejecutan, el problema puede radicar en

que si queremos que ciertas instrucciones se ejecuten constantemente, estas debe de incluirse dentro de una instrucción “por siempre” o repetir hasta que se cumpla.



### Buenas prácticas de Programación

- **NOMBRADO DE OBJETOS SIGNIFICATIVOS.** Nombrar a los objetos cuando los creamos con nombres significativos y no dejar el Objeto1..., que asigna por defecto Scratch.
- **UTILIZAR COMENTARIOS.** Para ayudar a otros programadores.
- **DESARROLLO ITERATIVO E INCREMENTAL.** Se divide el proyecto en pequeños objetivos. No se para al siguiente objetivo hasta que el anterior no funciona correctamente.
- **PRIMERO LA PROGRAMACIÓN ... LUEGO EL DISEÑO.** Se debe empezar por la programación y luego mejorar el diseño.



Para acceder a los distintos tutoriales.

Desde el siguiente código QR podrás acceder a todos los videos tutoriales realizados por Aprendizaje digital.



## Contenido

1.- Programación sin código.....	1
Versiones Scratch .....	1
Acceso a la Plataforma .....	2
Interface de Scratch .....	7
Nuestro primer programa en Scratch .....	12
2.- Por qué programar.....	14
¿Por qué no Programar? .....	14
¿Por qué Programar? .....	14
Conclusión .....	14
3.- Introducción a la programación.....	15
Pensamiento computacional.....	15
Algorítmicos .....	15
Programa informático .....	16
Lenguaje de Programación.....	16
Elementos básicos de la programación.....	18
4.- Objetos y Escenarios: APARIENCIA .....	21
Interface de nuestro programa en Scratch .....	21
OBJETOS/PERSONAJES .....	21
Programar la Apariencia Objetos/Personajes.....	26
FONDOS/ESCENARIO.....	28
Programar la Apariencia: Fondos/Escenario.....	29
5.- Estructura Básica Programación .....	30
Eventos.....	30
Mensaje entre objetos .....	30
Estructuras condicionales .....	31
Estructuras repetitivas .....	31
6.- Movimientos de Personaje Usuario.....	32
Posicionar objetos en la pantalla .....	32
Movimientos objetos controlado por el usuario .....	33
7.- Movimiento de Objetos solos .....	37
Movimientos automáticos de un Objeto: Pelota.....	37
Movimientos automáticos de un Objeto: Gravedad.....	37
Lanzamiento de un objeto por otro objeto: Proyectil .....	38
8.- Movimiento Escenario SCROLL .....	39
Scroll Horizontal de Fondo .....	39

Scroll Vertical de Fondo .....	42
9.- Contacto Personajes COLISIONES .....	43
Concepto de Colisión.....	43
Programar una colisión en Scratch.....	43
10.- Guardar datos: VARIABLES.....	44
Concepto de Variable .....	44
Variable en Scratch.....	44
Barras de puntuación y dibujos con vidas.....	46
11.- Sonido y Música .....	48
Sonido.....	48
Música .....	50
12.- Cámara y Micrófono .....	52
Cámara .....	52
Micrófono.....	53
13.- Crear nuevos Bloques: FUNCIONES .....	55
Concepto de Función .....	55
Programar Funciones en Scratch .....	55
14.- Copias de un Objeto: CLONES .....	57
Concepto de Clones.....	57
Programar Clones en Scratch .....	57
15.- Matemáticas y Texto.....	60
Operaciones Matemáticas en Scratch .....	60
Operaciones con Texto en Scratch.....	61
16.- Otros sensores .....	62
Otras instrucciones con sensores.....	62
17.- Text a Voz y Traducciones .....	64
Texto a Voz.....	64
Traducciones .....	65
18.- Dibujando con Scratch: LÁPIZ .....	66
Concepto de Lápiz en Scratch .....	66
Programar con el Lápiz en Scratch .....	66
19.- Explorar, Buscar y Compartir Proyectos .....	70
Explorar Proyectos .....	70
Compartir Proyectos .....	70
20.- Depuración de Programas: ERRORES.....	72
Errores en los programas .....	72

Errores típicos en Scratch.....	72
Buenas prácticas de Programación .....	73
Para acceder a los distintos tutoriales. ....	74