

---

# Table of Contents

Introducción	1.1
--------------	-----

---

## 1. Gamificación

Gamificación	2.1
Scratch	2.2
La idea del juego	2.3
Movimiento de la letra	2.4
Movimiento del lápiz	2.5
Colisión del lápiz con la letra	2.6
Resultado final	2.7

---

## 2. Gamificación (cont.)

Gamificación (cont.)	3.1
La idea del juego ampliada	3.2
Movimiento de la letra	3.3
Movimiento del lápiz	3.4
Movimiento del disparo	3.5
Colisión del disparo con un objeto	3.6
Últimos ajustes	3.7
Resultado final	3.8

---

## 3. Abriendo posibilidades

Abriendo posibilidades	4.1
Makey Makey	4.2
¿Qué es?	4.2.1
Plastilina I	4.2.2
Plastilina II	4.2.3
Piano	4.2.4
El cuerpo humano	4.2.5
Cuelga experiencias	4.2.6
ProgramoErgoSum	4.3
Vocales	4.3.1
Clonar Matar pájaros	4.3.2
2 jugadores Fútbol	4.3.3
Bloques Duck Hunt	4.3.4
Flappy movientos	4.3.5
Pong	4.3.6

---

---

<a href="#">Rebotes - Arkanoid</a>	4.3.7
<a href="#">Reinventar</a>	4.4
<a href="#">¿Por qué reinventar?</a>	4.4.1
<a href="#">Conviertete en un Scratcher</a>	4.4.2
<a href="#">Scratch para docentes</a>	4.5
<a href="#">Abrir cuenta como profesor</a>	4.5.1
<a href="#">Clase con Scratch</a>	4.5.2
<a href="#">Evaluar</a>	4.5.3
<a href="#">Consejos</a>	4.6
<a href="#">Retos</a>	4.7
<a href="#">Créditos</a>	4.8

---

## Introducción

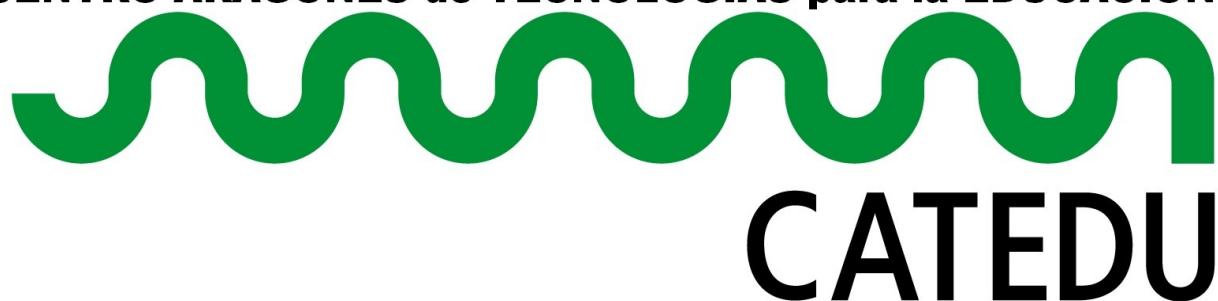
Este curso es **UNA CONTINUACIÓN DEL CURSO BÁSICO DE SCRATCH** por lo que precisa de sus conceptos básicos.

## Objetivos

- Seguir con la gamificación en Scratch: Aprender jugando con proyectos e introducir instrucciones más complejas.
- Conocer Makey Makey y sus posibilidades.
- Concepto de reinventar en Scratch
- Herramientas para evaluar en Scratch



**CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN**



## RoboTICa

Oferta de formación en Pensamiento computacional del Centro Aragonés de Tecnologías para la Educación.



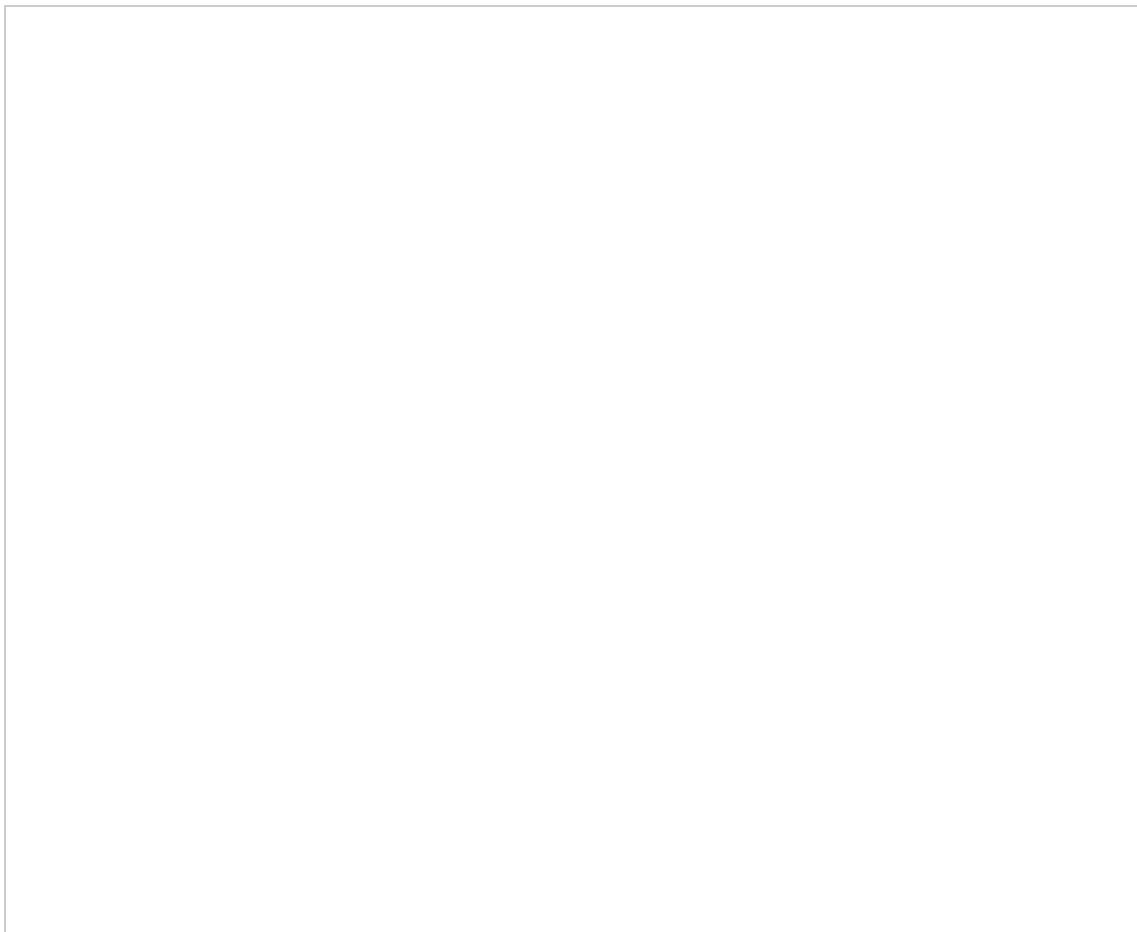


## Gamificación

Con los contenidos vistos hasta ahora [curso de Scratch básico](#), ya se ha tratado todo lo necesario para poder superar los objetivos de aprendizaje. En este módulo vamos a afianzar los conocimientos adquiridos. Para ello, vamos a crear un nuevo juego sencillo en Scratch.

Mira esta presentación, fíjate en los siguientes conceptos:

- Preguntas y respuestas
- Apuntar y pulsar
- Arcade
- Movimiento, rebote y colisiones
- Aventuras gráficas

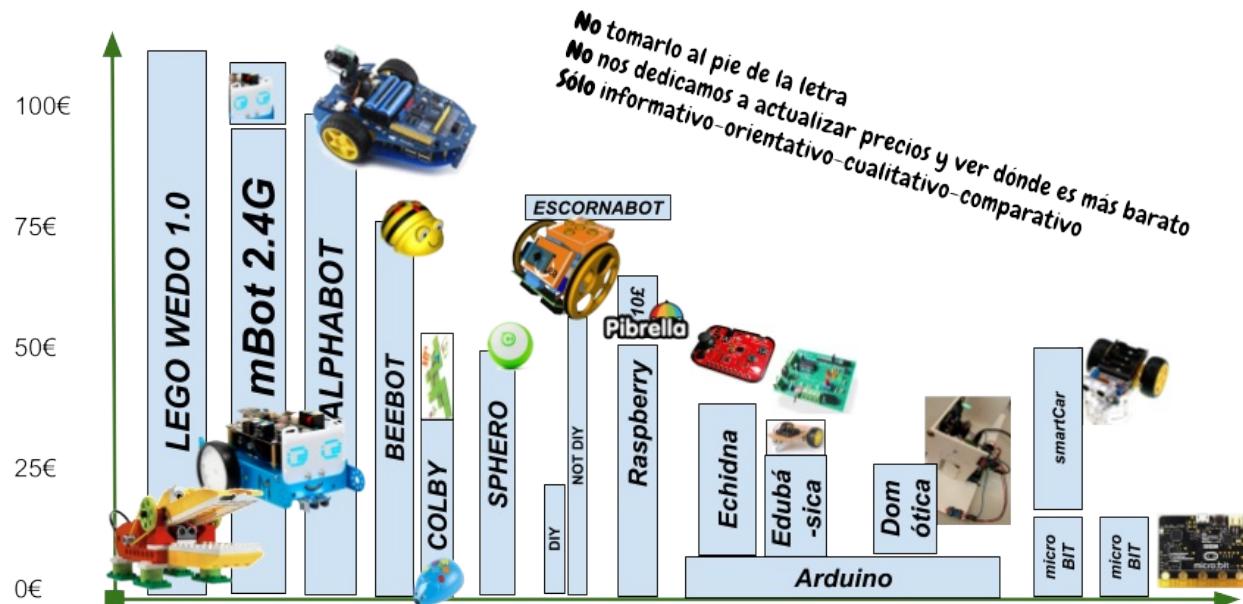


## Scratch

Antes de nada ¿Dónde se ubica **Scratch** dentro de las estrategias de pensamiento computacional? En CATEDU hemos elaborado la siguiente hoja de ruta de toda la oferta que tenemos para el pensamiento computacional:



### *Coste aproximado de los packs de robótica*



"No esperamos que todos los niños que aprenden a escribir se conviertan en novelistas, ni que todos los que estudian álgebra acaben siendo matemáticos; sin embargo, ambas son consideradas como habilidades fundamentales que todos los niños deben aprender. La programación informática también debería serlo",

Hadi Partovi cofundador de [Code.org](https://code.org), una organización sin ánimo de lucro que promueve la enseñanza de código en las escuelas en EE UU.

## Importante

Para hacer este curso hay saber algunos conocimientos de Scratch ([Curso Scratch básico de Aularagón por ejemplo](#)) por si acaso, contesta estas preguntas:

**Dado los siguientes bloques,  
¿Si pulso la tecla A se mueve 10 pasos?**

al presionar tecla any enviar message1  
al recibir message1 mover 10 pasos  
al presionar A mover 10 pasos

**SI      NO**

Programa Ergo Sum

@ProgramaErgoSum

Lecturas

[Video link](#)

## La idea del juego

Vamos a programar un juego con Scratch.

**La idea de este juego es lanzar un lapicero contra una letra con el objetivo de acertar a darle. La letra se moverá para evitar ser alcanzada. Cuando lo consigamos, la letra desaparecerá, y sumaremos la puntuación de 1 punto.**

Empezaremos montando los objetos que participarán en nuestro juego. Scratch tiene en su biblioteca de imágenes: letras y un lapicero. Con estas imágenes ya tenemos lo necesario para montar nuestros objetos.

Sin embargo, con idea de hacer el juego algo más original, podemos coger dibujos de Internet para montar el juego, o incluso crear con alguna herramienta de dibujo nuestros propios objetos. Para el juego que vamos a trabajar, vamos a utilizar los siguientes dibujos.

**Descárgatelos a tu ordenador (botón derecho - guardar imagen como) porque vamos a montar el juego con ellos.**



Creación propia, utilizando un editor de textos (OpenOffice Writer), simplemente añadiendo un rectángulo y una letra dentro, y luego capturando la pantalla.



Fuente: <https://openclipart.org/detail/64429/pencil> El lapicero está rotado con Gimp.



Fuente: <https://openclipart.org/detail/49363/blackboard>

## Caso práctico: Incluir los objetos de nuestro juego

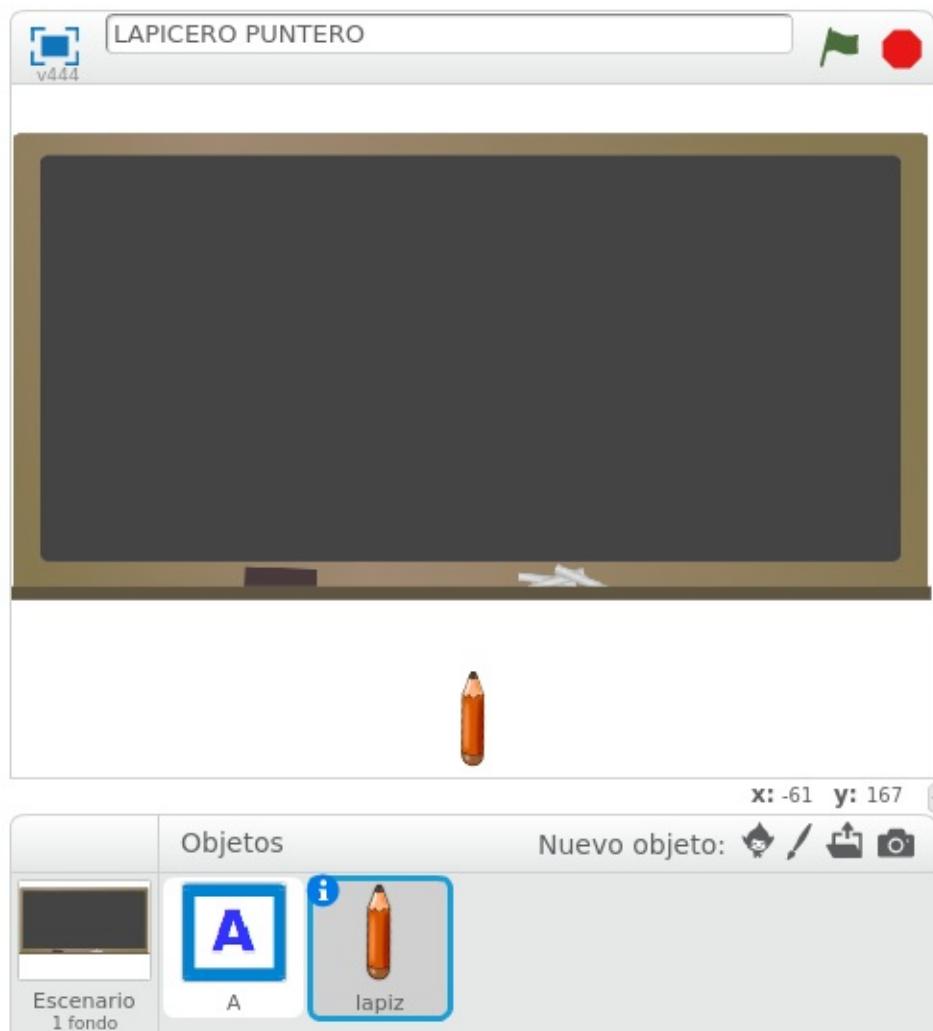
Empezamos a montar el entorno de objetos de nuestro juego.

Añade la letra A y el lápiz como objetos de Scratch. Revisa que el nombre del objeto A se llame A (botón dcho sobre el objeto - info) porque lo usaremos posteriormente.

Añade la pizarra como fondo.

Ponle un nombre al proyecto: LAPICERO PUNTERO

Solución



# Movimiento de la letra

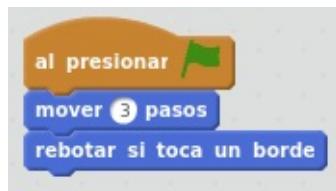
## Caso práctico: La letra se mueve sola

Vamos a hacer que la letra A se mueva. La letra A debe moverse de izquierda a derecha cambiando de dirección cuando llegue al borde.

1. Lo primero de todo: haz clic en el objeto de la letra A, para añadir su programa de bloques.
2. Empezaremos a arrancar las acciones al presionar Bandera (bloque Eventos: "al presionar bandera").
3. Haremos que se mueva 3 pasos (bloque Movimiento: "mover 3 pasos").
4. Cuando llegue al borde de la pantalla, haremos que cambie de dirección (bloque Movimiento: "rebotar si toca un borde").

Solución

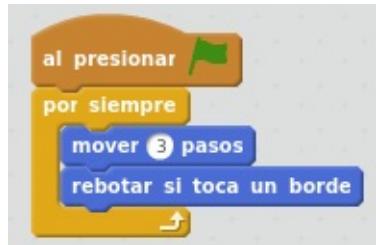
Empezamos a construir la solución. Empezamos poniendo estos bloques:



Al arrancar el programa haciendo clic en la Bandera, veremos que la letra A se mueve 3 pasos a la derecha, pero sólo se mueve una vez.

Por lo tanto, a los bloques que ya tenemos hay que **añadirles** un bloque de Control: "por siempre", y meter ahí dentro el movimiento y control de rebote. Esto genera un bucle, de forma que la letra A no deje de moverse nunca.

Siguiente paso: a los bloques que ya tenemos, les metemos el bloque "por siempre", y el programa quedará de la siguiente forma:



Para asegurarnos que la letra A está posicionada siempre en la misma posición al iniciar el juego, ponemos su posición fija al inicio, por ejemplo con  $x=-160$   $y=110$  (bloque Movimiento: "ir a x: -160 y: 110").

Hacemos que A apunte a la derecha para que empiece a moverse hacia la derecha (bloque Movimiento: "apuntar en dirección 90°"). Es posible que el programa funcione bien incluso sin este bloque, pero así nos aseguramos que la letra se va a empezar a mover hacia la derecha y no hacia otro lado.

Con esto ya hemos terminado el PROGRAMA que hace que la letra se mueva sola, y siempre empiece desde la misma posición.





## Movimiento del lápiz

### Caso práctico: mover el lápiz

Haremos que el lápiz se mueva a izquierda y derecha al presionar flechas izquierda y derecha.

Solución



### Caso práctico: lanzar el lápiz

Al presionar Flecha arriba, haremos que el lápiz se mueva constantemente hacia arriba, hasta llegar al borde de arriba.

- Iniciamos el programa con el bloque de Control "al presionar tecla flecha arriba".
- Utilizar el bloque Movimiento "cambiar y por 10" para hacer que el lápiz se mueva hacia arriba.
- Añadir el bloque de Control "repetir hasta que < >" para hacer que el bucle se repita constantemente, y que se salga del bucle al llegar a una condición. Habrá que meter el bloque de Movimiento dentro del bucle.
- Utilizar el bloque Sensor "tocando.. borde" para detectar que hemos llegado hasta arriba. Esta será la condición de salida del bucle.

Solución



Cuando llegue el lápiz arriba del todo, es decir: cuando se termine el bucle que comprueba si el lápiz ha tocado un borde, lo volvemos a poner en su situación inicial.

Solución



## Colisión del lápiz con la letra

### Caso práctico: detectar colisión con la letra

El objetivo es alcanzar con el lápiz la letra que se mueve.

Para controlar la colisión seguimos añadiendo bloques en la programación del objeto lápiz, en el programa que se inicia "al presionar tecla flecha arriba".

Si detectamos que estamos tocando la letra A, enviaremos el mensaje "impacto-A" a la letra A para decirle que desaparezca.

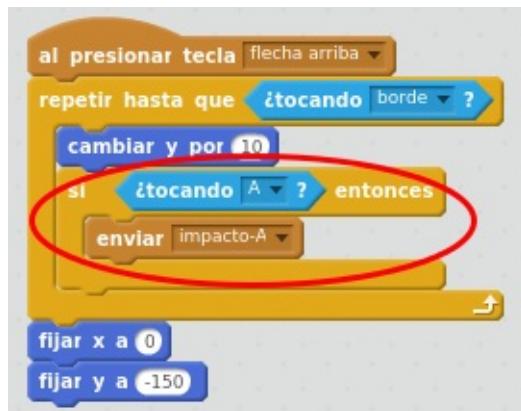
Habrá que pensar dónde colocar estos bloques.

Solución

Estos son los bloques que controlarán que si el lápiz toca el objeto A, entonces se envía el mensaje "impacto-A":



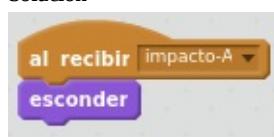
Estos bloques los colocaremos justo después de haber movido el lápiz en 10 posiciones hacia arriba:



Ahora tenemos que recoger el mensaje "impacto-A" en el objeto A. Por lo tanto, haremos clic en el objeto A para modificar los bloques de su programación.

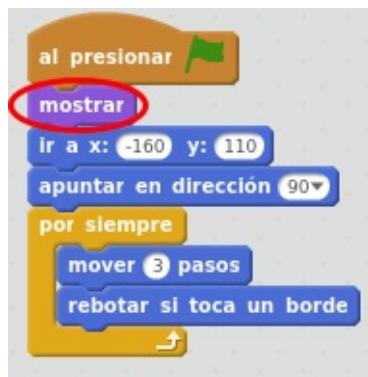
Añadiremos un bloque de programación en el objeto A: Cuando se reciba el mensaje "impacto-A", haremos que el objeto A desaparezca.

Solución



Ahora, cuando el objeto A es impactada por el lápiz, desaparece y ya no vuelve a aparecer. Tendremos que hacer que al inicio del juego, el objeto A aparezca visible. Lo podremos añadir en el programa ya existente que mueve la letra A.

Solución

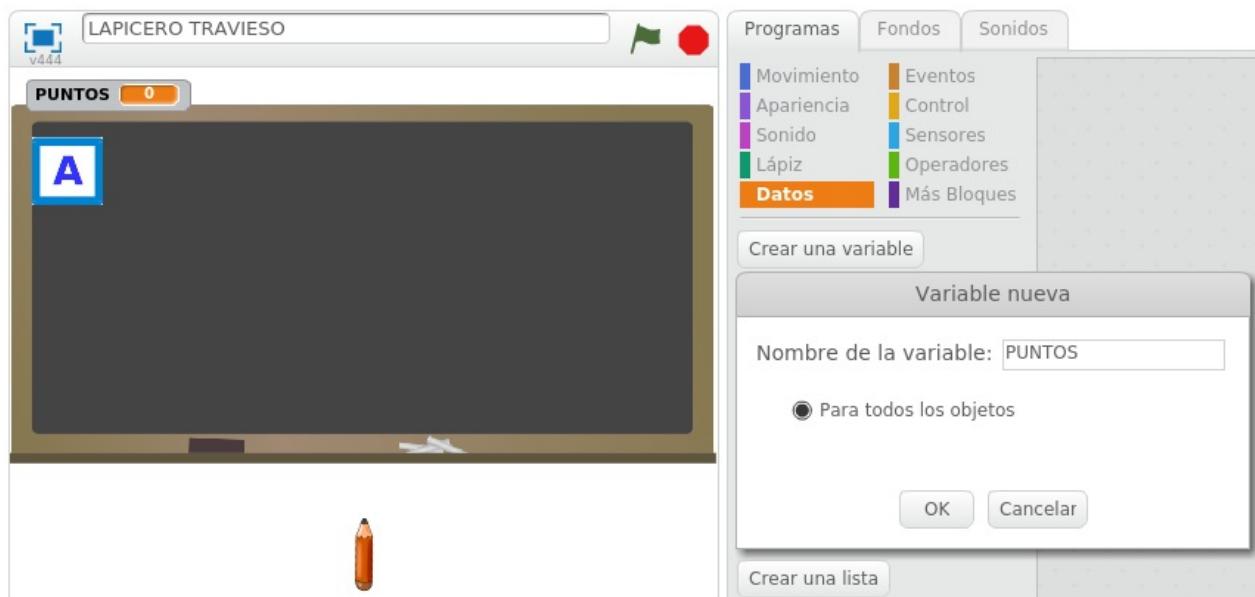


## Caso práctico: Añadir puntuación

Añadiremos un marcador de puntos al juego. Haremos que los puntos se incrementen al detectar la colisión del lápiz con la letra.

Crear una variable llamada PUNTOS.

Solución



En la programación del objeto A:

- Al inicio del programa, ponemos los puntos a 0.
- En el momento de detectar colisión, sumamos 1 en la variable puntos.

Solución





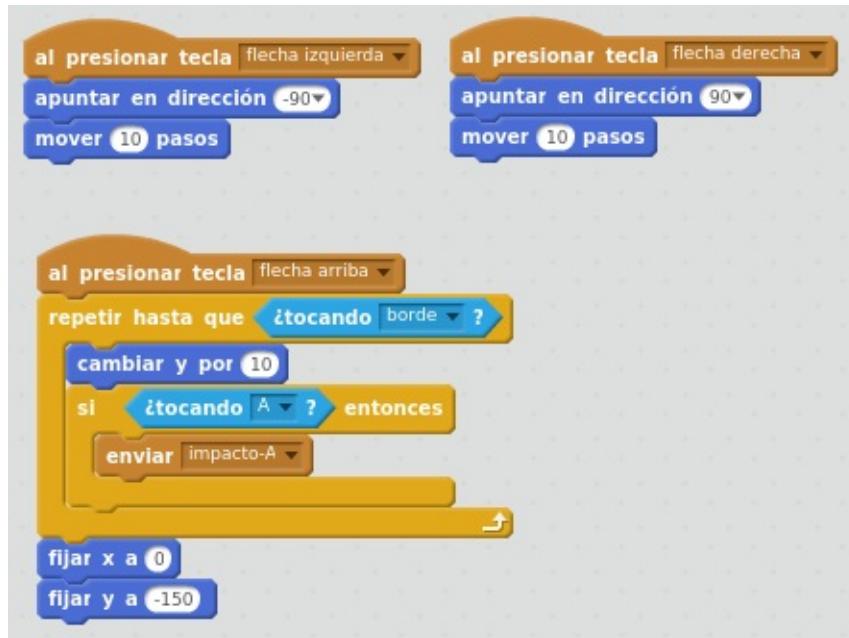
## Resultado final

Hasta aquí, hemos creado una programación para el juego de lanzar un lápiz contra una letra.

Los programas del objeto A quedan por tanto de la siguiente forma:



Los programas del objeto lapiz quedan por tanto de la siguiente forma:



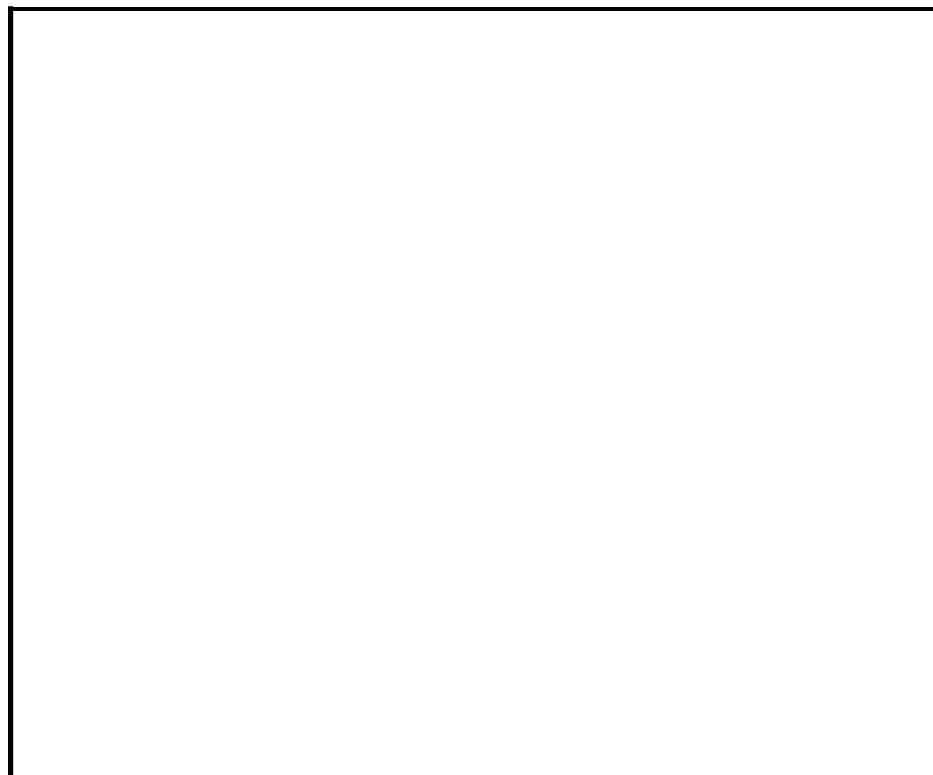
¡YALO TENEMOS!. Utilizando Pensamiento Computacional, hemos conseguido montar un pequeño juego con Scratch.

## Gamificación (cont.)

Es este bloque de contenidos se plantea un reto mayor: continuar mejorando el juego que ya tenemos creado. Va a requerir añadir más complejidad en los programas de los objetos, ¡pero nada que no esté ya a vuestro alcance! Espero que os sintáis con ganas de continuar con este bloque de contenidos que se sale de la programación planteada en el curso de Aularagon.

El resultado de lo que vamos a conseguir lo puedes ver aquí:

[Ver resultado final](#)



## La idea del juego ampliada

Vamos a programar un nuevo juego con Scratch, que va a ser una ampliación del anterior. La idea de este juego es la del típico matamarcianos. Pero en nuestro caso, lo que vamos a hacer es disparar pintura con un lápiz a unas letras. Las letras se moverán para evitar ser alcanzadas. El objetivo final será acertar a disparar a todas las letras.

Empezaremos montando los objetos que participarán en nuestro juego. Scratch tiene en su biblioteca de imágenes: letras, lapisero, rayo. Con estas tres imágenes ya tenemos lo necesario para montar nuestros objetos.

Sin embargo, con idea de hacer el juego algo más original, podemos coger dibujos de Internet para montar el juego, o incluso crear con alguna herramienta de dibujo nuestros propios objetos. Para el juego que vamos a trabajar, vamos a utilizar los siguientes dibujos. Descárgatelos a tu ordenador (botón derecho - guardar imagen como) porque vamos a montar el juego con ellos.



Creación propia, utilizando un editor de textos (OpenOffice Writer), simplemente añadiendo un rectángulo y una letra dentro, y luego capturando la pantalla.



Creación propia, utilizando un editor de textos (OpenOffice Writer), simplemente añadiendo un rectángulo y una letra dentro, y luego capturando la pantalla.



Creación propia, utilizando un editor de textos (OpenOffice Writer), simplemente añadiendo un rectángulo y una letra dentro, y luego capturando la pantalla.



Fuente: <https://openclipart.org/detail/29133/pencil> El lapisero está rotado con Gimp.



Creación propia, utilizando un programa de edición de imágenes (Gimp)



Fuente: <https://openclipart.org/detail/49363/blackboard>

## Caso práctico: incluir los objetos de nuestro juego

Empezamos a montar el entorno de objetos de nuestro juego.

Añade las letras A, B y C, el lápiz y el disparo como objetos de Scratch. Revisa que los nombres de objetos de las letras A, B y C se llamen a, b, y c, (botón dcho sobre el objeto - info) porque usaremos estos nombres posteriormente.

Añade la pizarra como fondo.

Ponle un nombre al proyecto: LAPICERO TRAVIESO

Solución

Te tienes que quedar así:



## Movimiento de la letra

### Caso práctico

Vamos a hacer que la letra A se mueva. La letra A debe moverse de izquierda a derecha cambiando de dirección cuando llegue al borde. Esto ya lo hemos hecho en el juego anterior.

Al presionar bandera, posicionamos la letra en un lugar fijo y apuntando a la derecha. A continuación haremos que, de forma indefinida, la letra se mueva 3 pasos, y cuando llegue al borde de la pantalla, haremos que cambie de dirección.

Solución



## Movimiento del lápiz

### Caso práctico: el lápiz se mueve con las teclas izquierda y derecha

Queremos que el lápiz se mueva a la izquierda cuando presionemos la flecha izquierda. Y que se mueva a la derecha cuando presionemos la flecha derecha.

Esto ya lo hemos hecho en el juego anterior. Sin embargo, si somos "exquisitos", habrás observado que si mantienes pulsada una tecla de flecha izquierda o derecha, debido a que hay un retardo en el teclado antes de empezar a mandar la señal de repetición a nuestro programa, se produce también un retardo en el primer movimiento del lápiz.

Vamos a construir una solución que evita este inconveniente con la siguiente lógica: Al presionar Bandera, estaremos comprobando de forma indefinida si hay alguna tecla presionada. Moveremos el lápiz 10 pasos a la derecha si se presiona la tecla "flecha derecha", y moveremos el lápiz 10 pasos a la izquierda si se presiona la tecla "flecha izquierda".

Empezamos a construir la solución paso a paso. En primer lugar, vamos a hacer que al presionar Bandera, haremos que si la tecla "flecha derecha" está presionada, mover el lápiz 10 pasos a la derecha

Solución



Pero esto no funciona, porque estos bloques sólo se arrancan una vez. Y necesitamos que el programa esté constantemente comprobando si la tecla está presionada. Por lo tanto, tenemos que poner todo dentro el bloque de Control "por siempre", para que el programa esté constantemente comprobando si la tecla está presionada. Ahora ya habremos conseguido que el lápiz se mueva a la derecha cuando presionemos la tecla de flecha derecha,

Solución



Lo único que queda es añadir la comprobación de si la tecla de flecha izquierda esté presionada, se mueva el lápiz 10 pasos a la izquierda.

Solución



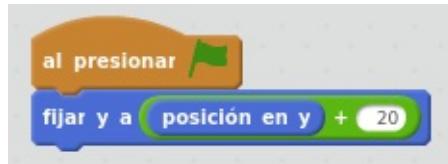
## Movimiento del disparo

### Caso práctico: disparar al presionar espacio

Vamos a hacer que cuando se presione la tecla espacio, el disparo de pintura salga del lápiz y empiece a moverse hacia arriba hasta que llegue al borde superior.

Empezamos haciendo que el disparo se mueva en 20 pasos hacia arriba: Fijaremos la posición Y del objeto disparo sumándole 20 a su posición actual. Utilizaremos el bloque de Operadores "+".

Solución



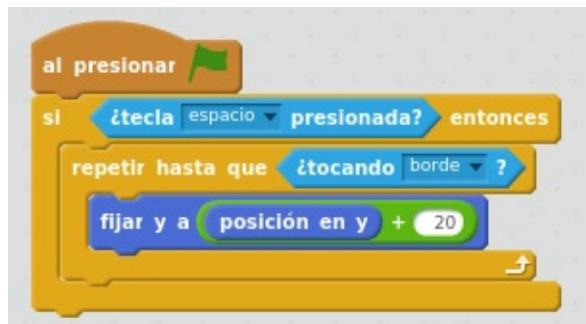
Ahora, el disparo tiene que moverse 20 pasos hacia arriba pero constantemente, hasta que llegue al borde.

Solución



Ponemos como condición previa a todo esto que se ponga en marcha si se ha presionado la tecla espacio.

Solución



Esto sólo se arranca una vez, necesitamos meter un bucle para que el

programa esté constantemente comprobando si se ha presionado la tecla espacio:



El disparo debe salir del lápiz, por tanto, debemos posicionar el disparo encima del lapicero. posicionaremos el disparo 50 pixeles por encima de la posición del lápiz. ¿En qué momento hay que posicionarlo?: antes del bucle que mueve el disparo hacia arriba.

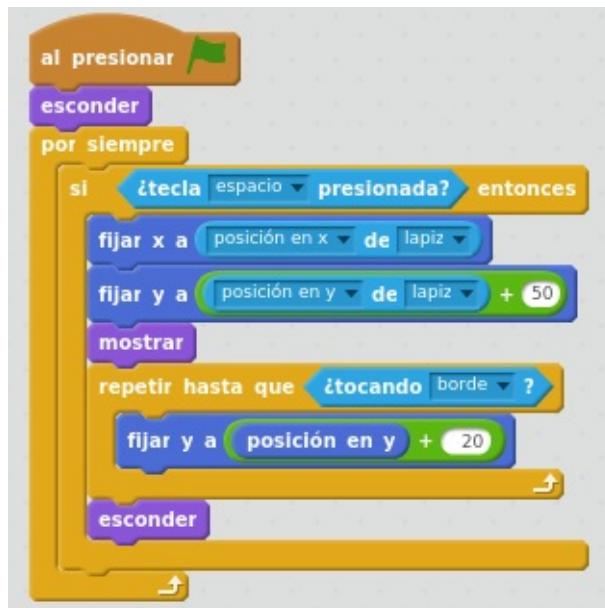
Solución



¿Cuándo mostrar y ocultar el disparo?

- Al principio del todo, debemos ocultar el disparo.
- Una vez que tenemos el disparo posicionado encima del lapicero, lo mostramos
- En última instancia, hacemos que desaparezca después de detectar que ha tocado el borde.

Solución



## Colisión del disparo con un objeto

### Caso práctico: control de la colisión del disparo con una letra

Prueba

La colisión la vamos a controlar dentro del movimiento del disparo, por lo que continuaremos añadiendo más programación al objeto de disparo. También necesitaremos añadir más programación en las letras. El objetivo es que mientras se está moviendo el disparo, comprobar si colisiona con alguno de los objetos (a, b, c). Si colisiona, quitaremos la letra de la escena, para que no moleste, y sumaremos ¡1 punto al marcador!

Dentro del objeto disparo, si el objeto disparo toca al objeto A, enviaremos el mensaje "impacto-A". Este mensaje habrá que crearlo en el bloque Eventos - "enviar..." - nuevo mensaje.

Solución

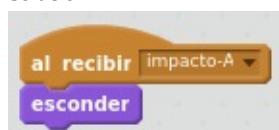


¿Dónde colocaremos estos bloques de programación?: en la programación del objeto disparo, justo después de haberlo movido 20 pasos hacia arriba:



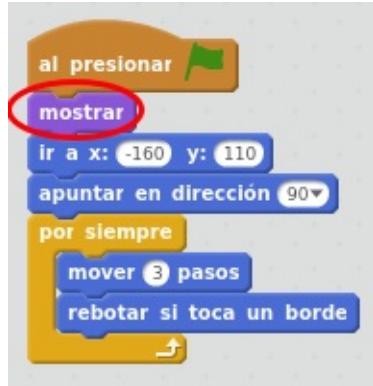
En la programación del objeto A, necesitaremos recoger el mensaje "impacto-A" en el objeto A. Una vez recogido el mensaje, ocultaremos el objeto A.

Solución



Ahora resulta que cuando el objeto A es impactado, se oculta y ya no se muestra ni aunque iniciemos una nueva partida. La solución es hacer que al inicio del juego, el objeto A se muestre.

Solución

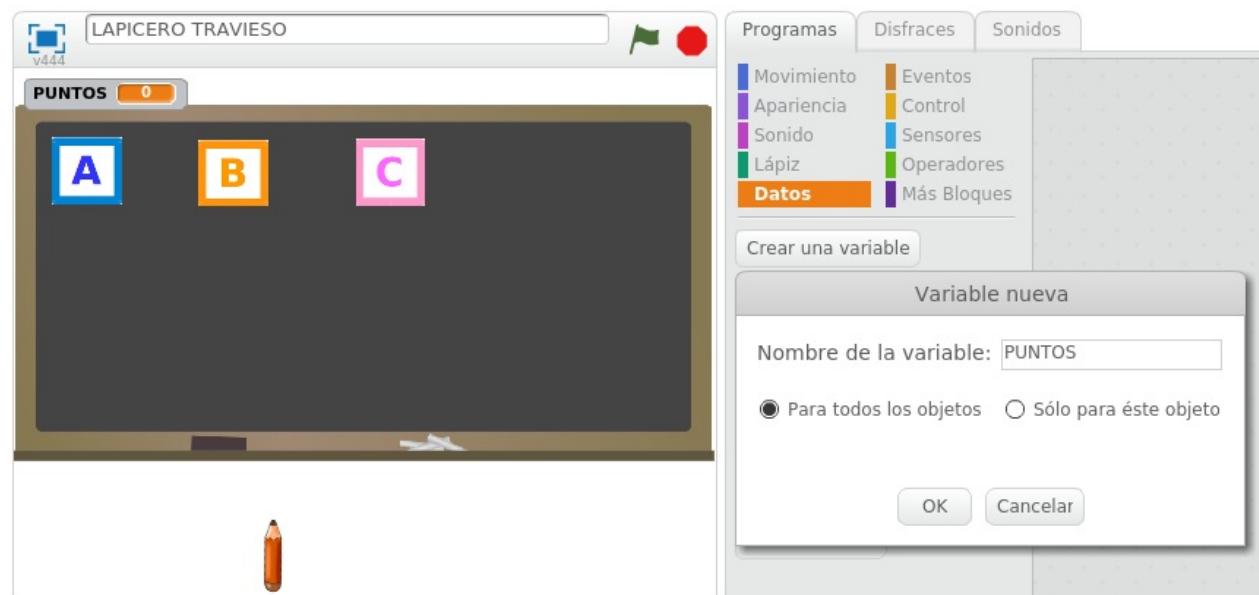


## Caso práctico: ¡Ahora hay que sumar los puntos del juego!

Tendremos los puntos de juego visibles en la pantalla. Haremos que al colisionar el disparo con un objeto, se sume 1 punto.

Hay que añadir una variable de puntos. Al añadirla, la variable se verá en el escenario de juego, la posicionaremos en la zona superior izquierda.

Solución



%/accordion% En la programación del objeto A, cada vez que se detecte que el objeto A ha sido colisionado por el disparo, se sumará 1 punto en la puntuación del juego.

Solución



## Últimos ajustes

### Caso práctico: Programar las letras B y C

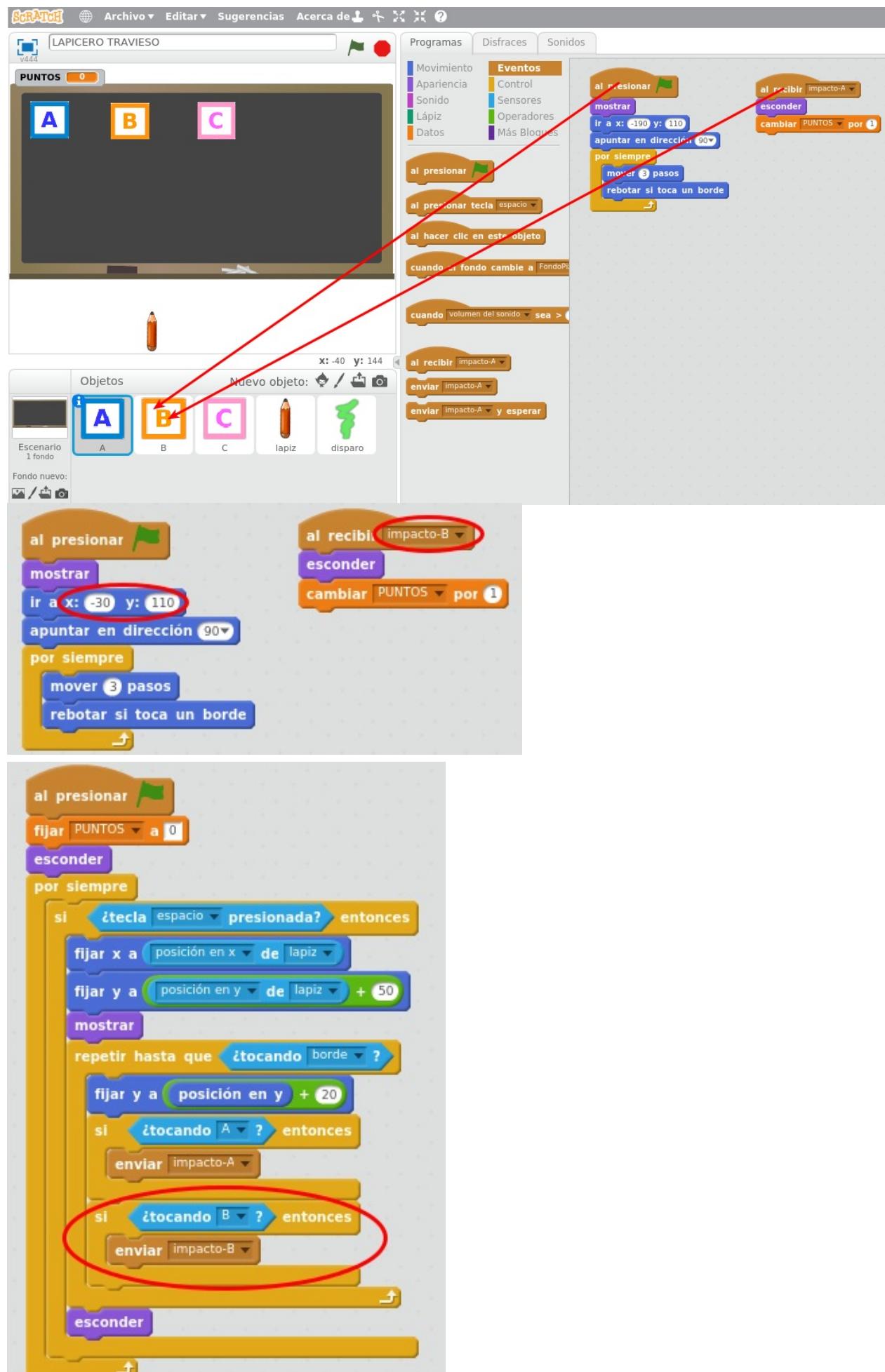
Ahora hay que programar las letras B y C igual que la A para que tenga el mismo comportamiento. Su programación será prácticamente igual a la del objeto A, cambiando el nombre del mensaje y sus coordenadas.

- Para no empezar de cero, es posible arrastrar todos los bloques de programación de la A sobre el objeto B.
- Posteriormente, hay que cambiar datos específicos para objeto B:
  - El mensaje será "impacto-B"
  - Las coordenadas iniciales para B serán x=0 y=110

De igual forma a lo realizado en la letra B, replicamos la programación para el objeto C.

- Su mensaje será "impacto-C"
- La posición fija del objeto C puede ser x=160 y=110

Solución



**OTRA OPCIÓN:** Podríamos duplicar el objeto A. Posteriormente

habría que:

- Añadirle el disfraz de letra B
- Borrarle al objeto el disfraz de letra A.
- Cambiarle el nombre al objeto por "A" (Botón dcho del ratón sobre el objeto B - info)
- Cambiar los datos específicos del objeto B (mensaje "impacto-B", coordenadas -30, 110), y no olvidar añadir la comprobación de colisión en la programación del disparo.

## Caso práctico: Añadir un temporizador

Añadamos un temporizador a nuestro juego.

Crear una variable llamada TIEMPO.

En la programación del Fondo (los fondos también pueden tener sus propios programas), incluiremos un nuevo programa. Al inicio reiniciaremos el cronómetro. En un bucle que se ejecutará siempre, pondremos el valor del cronómetro en la variable TIEMPO y así se visualizará en pantalla.

Solución



## Caso práctico: último reto, controlar el fin de juego

Cuando hayamos impactado con todas las letras, mostrar un mensaje en pantalla informando de "Fin de juego".

Una opción sería controlar constantemente el valor de los puntos, y hacer que cuando lleguen a 3, se acabe el juego. Pero otra posible solución algo mejor pensada puede ser controlar en una lista las letras que han sido impactadas. Para ello:

En la programación del lapicero, añadir un nuevo programa, que al iniciar el juego esté comprobando constantemente si hemos impactado las tres letras, y en tal caso mostraremos el mensaje "Fin de juego". Cómo podemos hacerlo:

- Añadiremos un disfraz nuevo al objeto lapiz. Construimos nosotros mismos el disfraz que va a ser el texto "Fin de juego".
- Crear una lista (pestaña Programas - Datos - Crear una lista, que podemos llamar LETRAS).
- Añadimos un bloque de programa en el objeto lapiz, que se inicie al presionar Bandera, donde de forma indefinida comprobaremos si la lista de letras contiene "a", "b" y "c". En tal caso, cambiaremos el disfraz del lapiz por el disfraz de "findejuego" y detendremos todos los programas.
- Tendremos que hacer que al inicio del juego, el disfraz del lapiz sea el del lapicero, para que no se quede fijo el mensaje de "Fin de juego". Y también habrá que ocultar y vaciar la lista.

En cada objeto de letra: añadir un valor a la lista cada vez que se impacte una letra.

En el objeto de lapiz: Crear una lista. Ocultarla y vaciarla al inicio del juego.

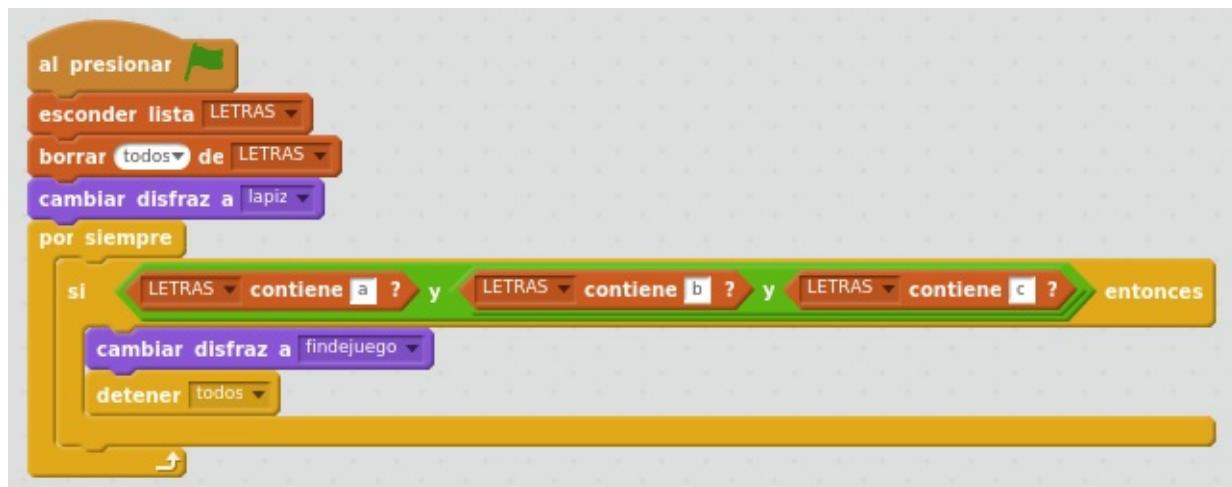
Solución

Crear una lista (pestaña Programas - Datos) y la llamamos LETRAS.

En la programación de las letras, añadir este bloque tras recibir el mensaje de impacto:



El programa del lápiz para controlar el fin de juego quedaría así:



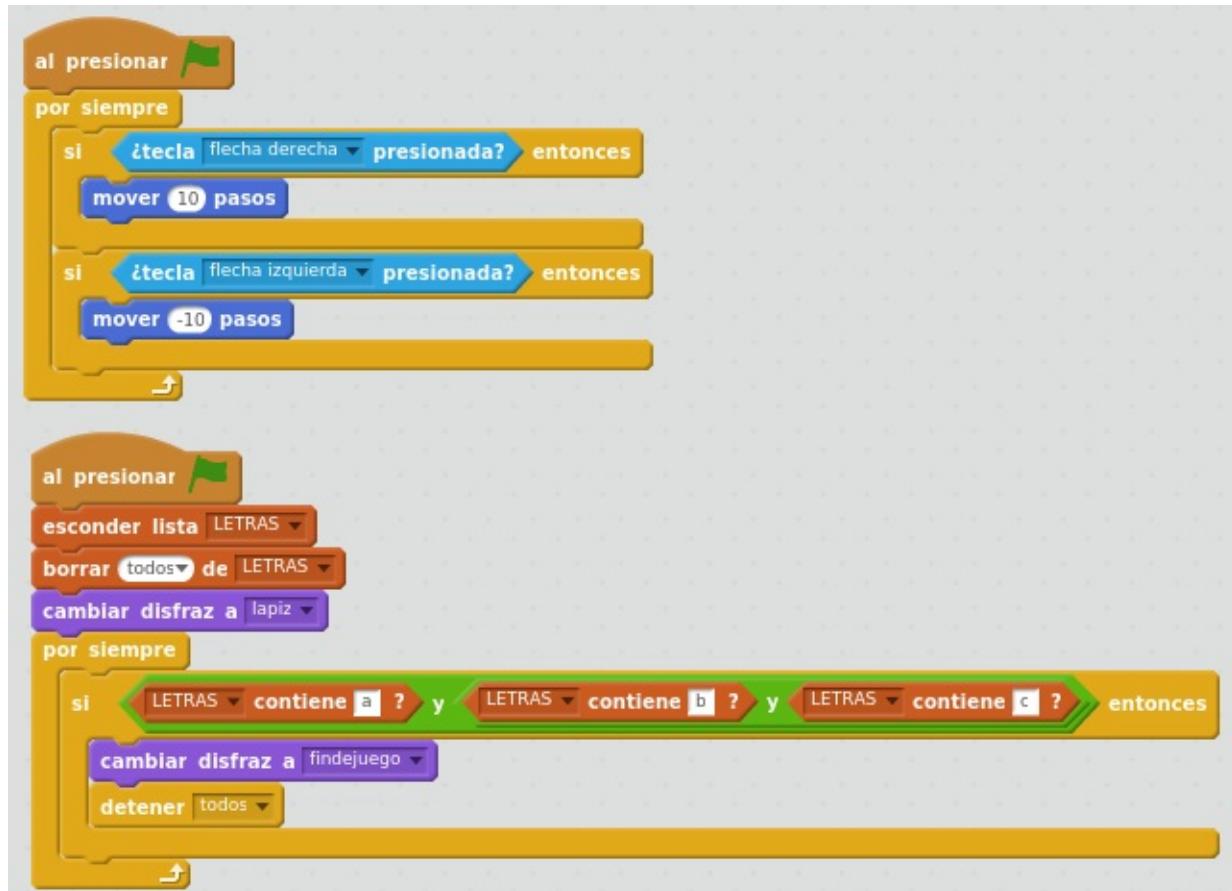
## Resultado final

Hemos creado una programación para el juego de lanzar pintura desde un lápiz contra varias letras.

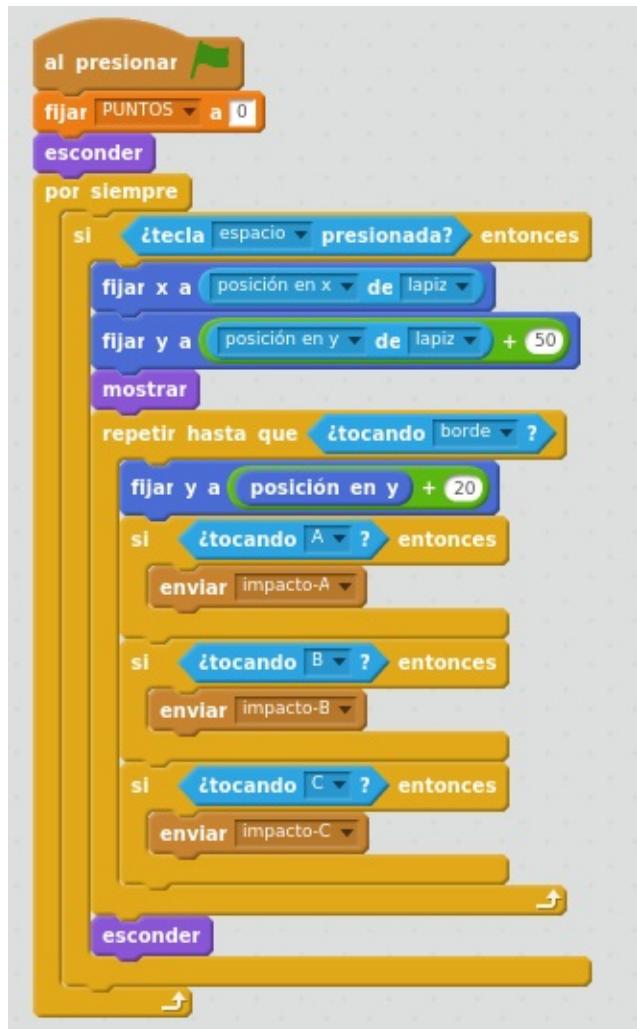
Los programas de cada letra, por ejemplo de la letra A, quedan de la siguiente forma:



Los programas del lápiz quedan:



El programa del disparo queda:

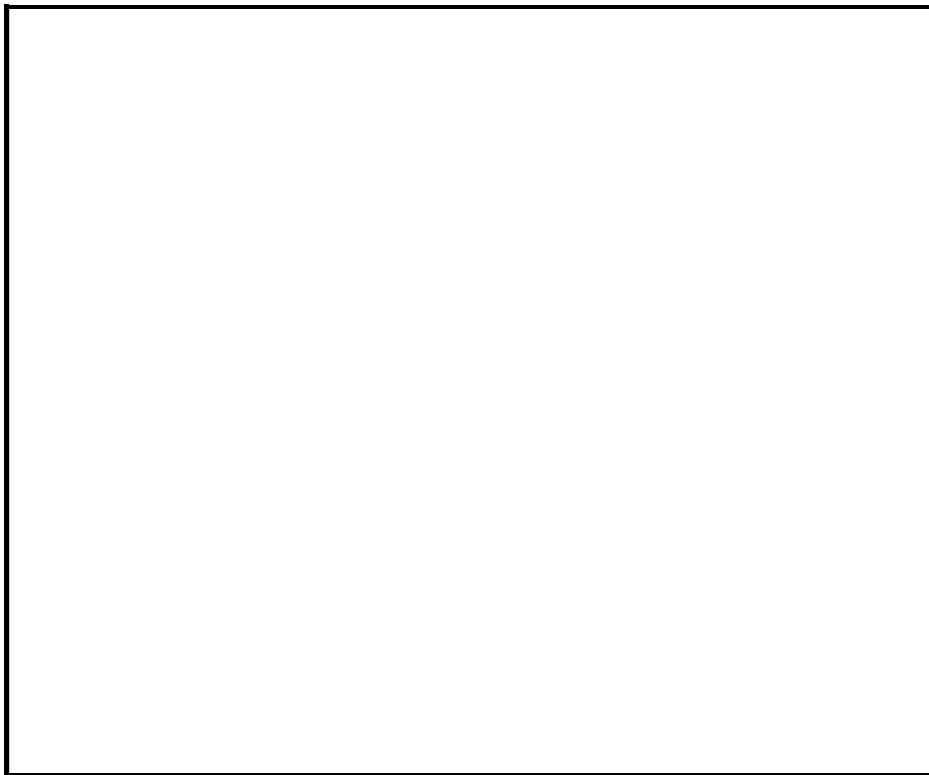


El programa de la imagen de fondo queda:



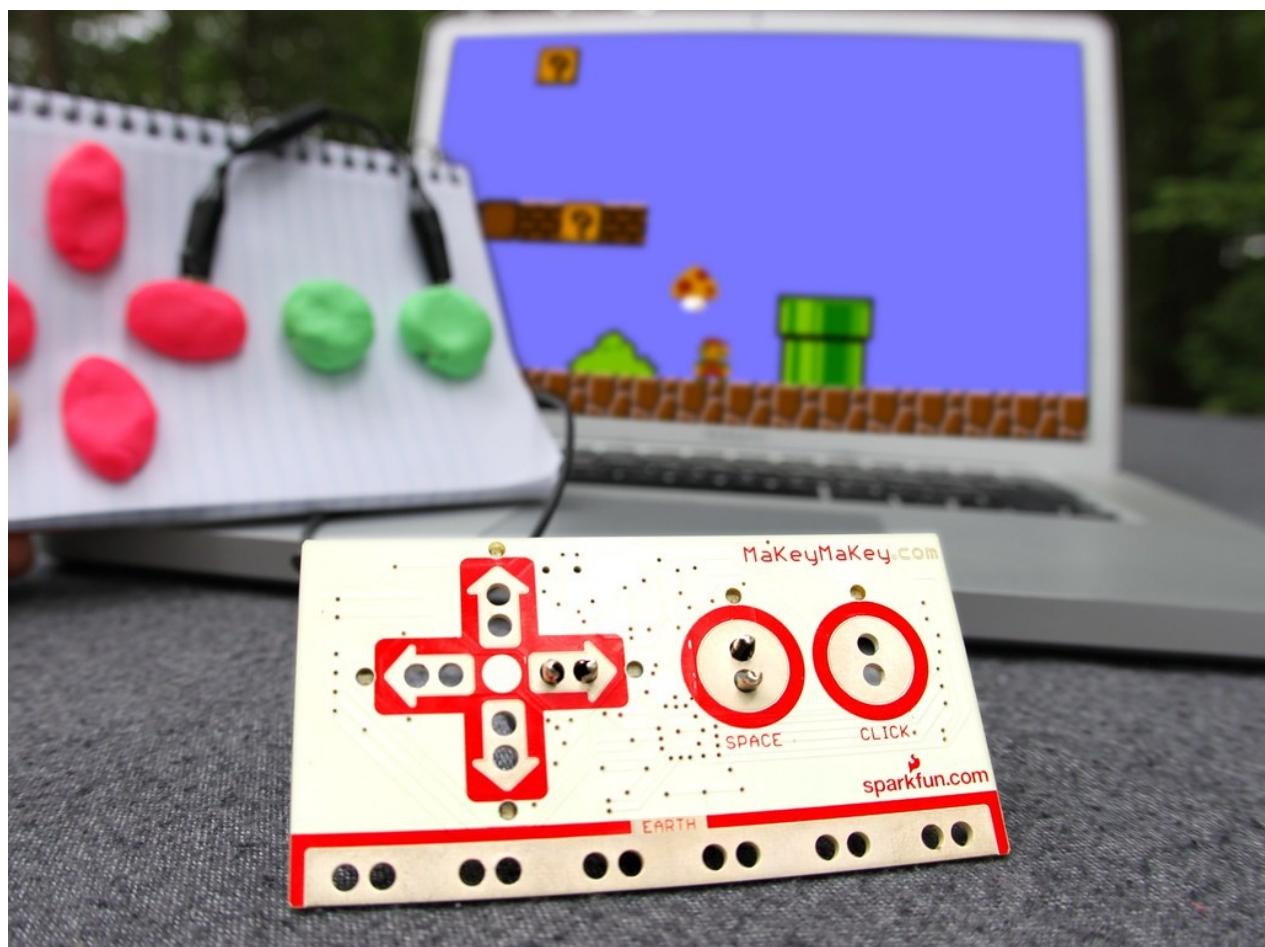
**¡YALO TENEMOS!. Utilizando Pensamiento Computacional, hemos conseguido montar un juego completo con Scratch.**

[Ver resultado final](#)

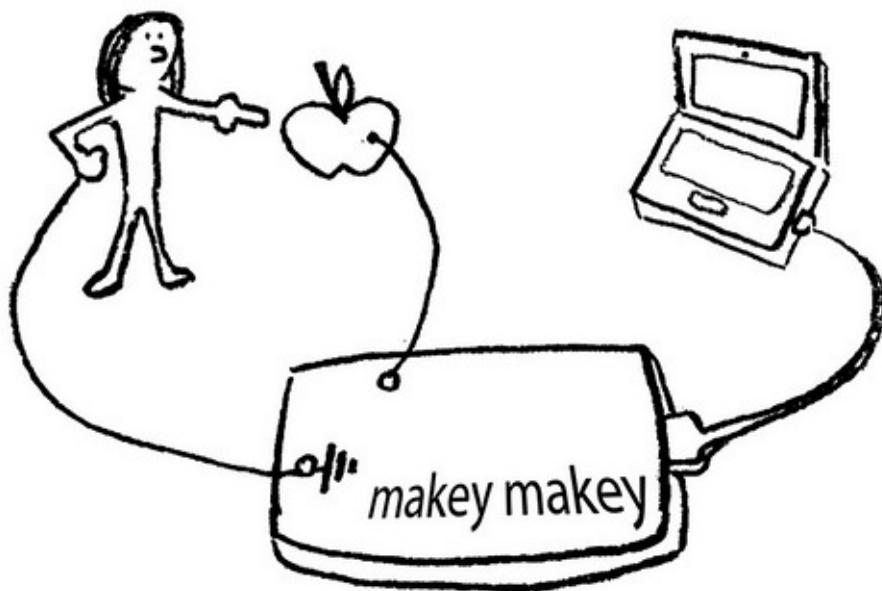


## Abriendo posibilidades

En este módulo, algunas secciones sólo funcionan con [Scratch on-line](#), evidentemente las que son de Internet.



## Makey Makey



Puedes pedir un Makey Makey para hacer este curso en [CATEDU-ROBOTICA](#)

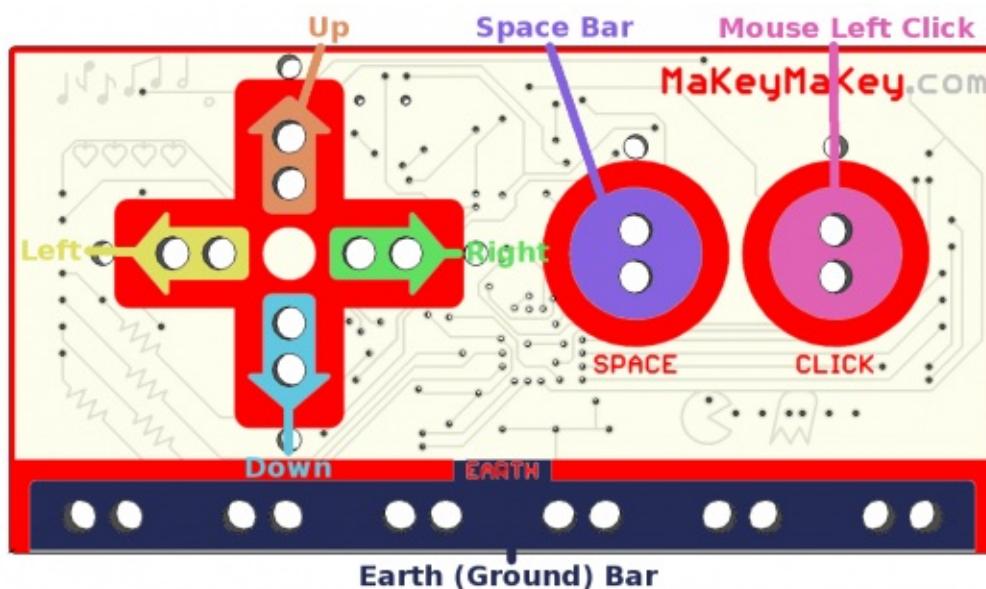
## ¿Qué es?

Es simplemente una entrada al ordenador por USB, como un teclado o ratón, pero por contacto:



[Video link](#)

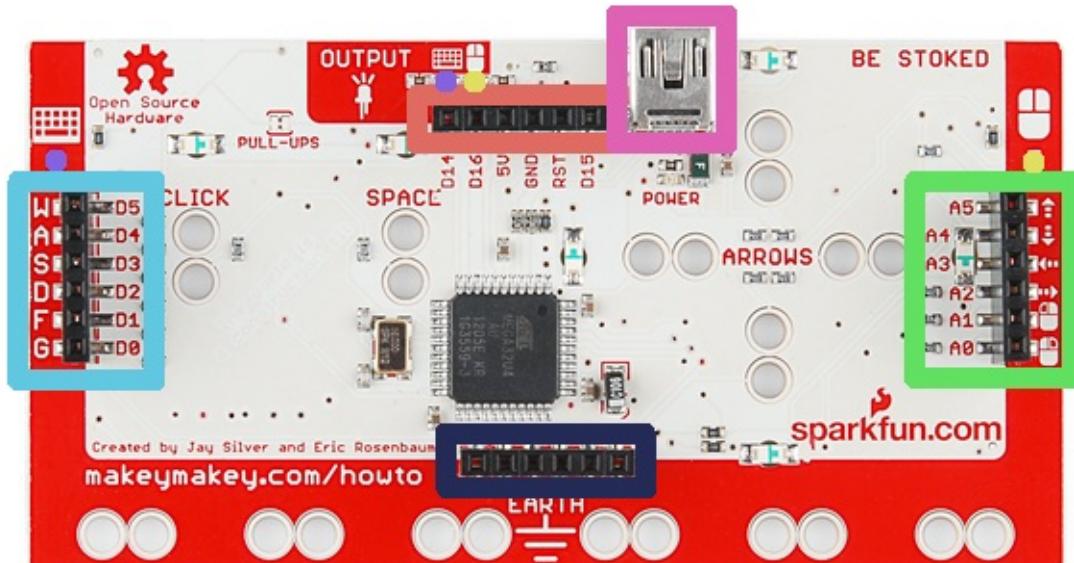
Las entradas principales son las flechas del teclado, la barra espaciadora y el click del ratón



Fuente: <https://learn.sparkfun.com/tutorials/makey-makey-quickstart-guide>

Debajo de la placa tiene acceso a

- las teclas **W, A, S, D, F, y G**
- **botón derecho, izquierdo del ratón y sus movimientos**



- USB Connector (Mini-B)
- Keyboard (W, A, S, D, F, G)
- Mouse (Up, Down, Left, Right, Left Click, Right Click)
- Keyboard LEDs
- Mouse LEDs
- Earth (Ground)
- Output/Expansion Header (D14, D16, 5V, GND, Reset, D15)

Fuente: <https://learn.sparkfun.com/tutorials/makey-makey-quickstart-guide>

Se pueden alterar estos parámetros entrando en su código y modificandolo [[+info](#)]

Cuesta alrededor de **70€**

## Plastilina I

### Propuesta Cuadro Musical

Puede ser un cuadro, o también un mapa, un panel gráfico con las partes del cuerpo, relieve ...



[Video link](#)

Solución

Para hacer la plastilina conductora aquí tienes una receta:

<http://www.comofuncionainternet.net/circuitos-con-plastilina/>

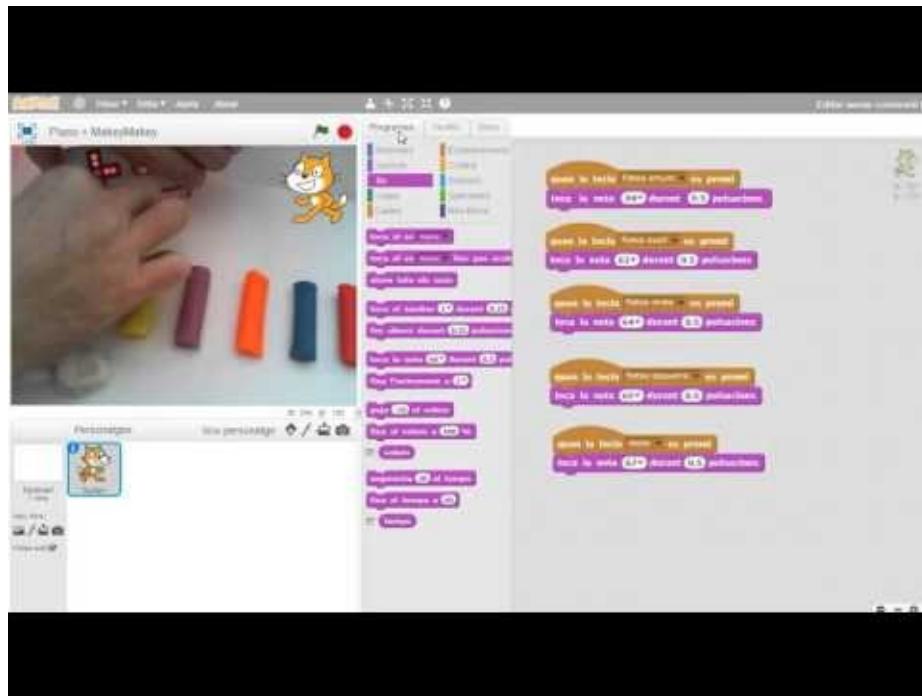
**Una pega: No dura nada, a los pocos días se endurece o se pudre y huele mal.**

Conectas la tecla que quieras con la música elegida, en este caso conecto la plastilina con el "a" del Makey y que suene Vivaldi



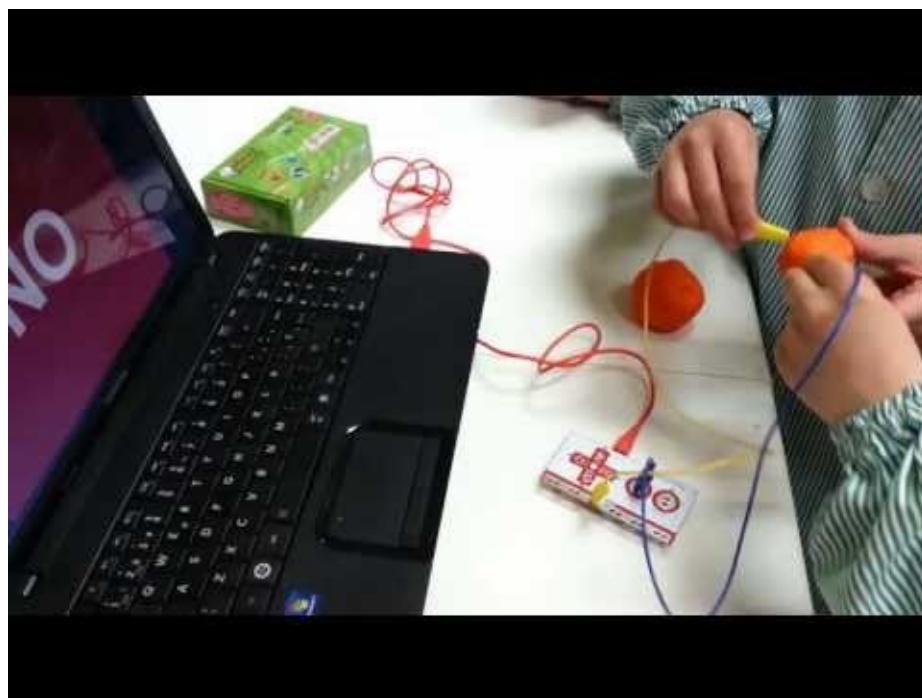
### Otras propuestas con plastilina

Piano



[Video link](#)

Conductor o no conductor



[Video link](#)

## Plastilina II

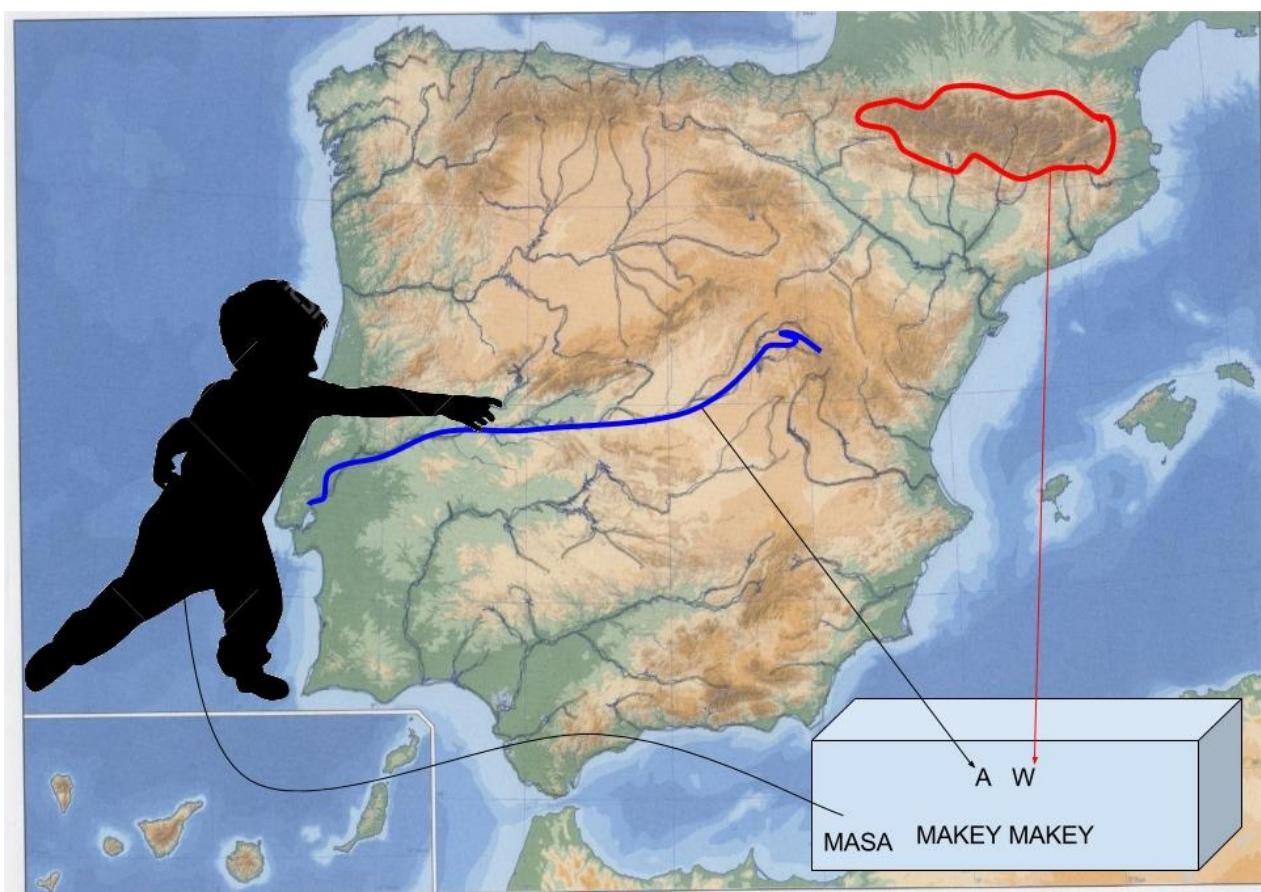
### Propuesta

Hacer un mapa de relieve interactivo con plastilina

Por ejemplo: Ponemos el mapa físico de la península y plastilina conductora:

- Una plastilina en **Pirineos** lo conectamos al **W** del Makey Makey
- Otra plastilina en el río Tajo lo conectamos al **A** del Makey Makey
- el niño tiene que tocar la masa del Makey Makey

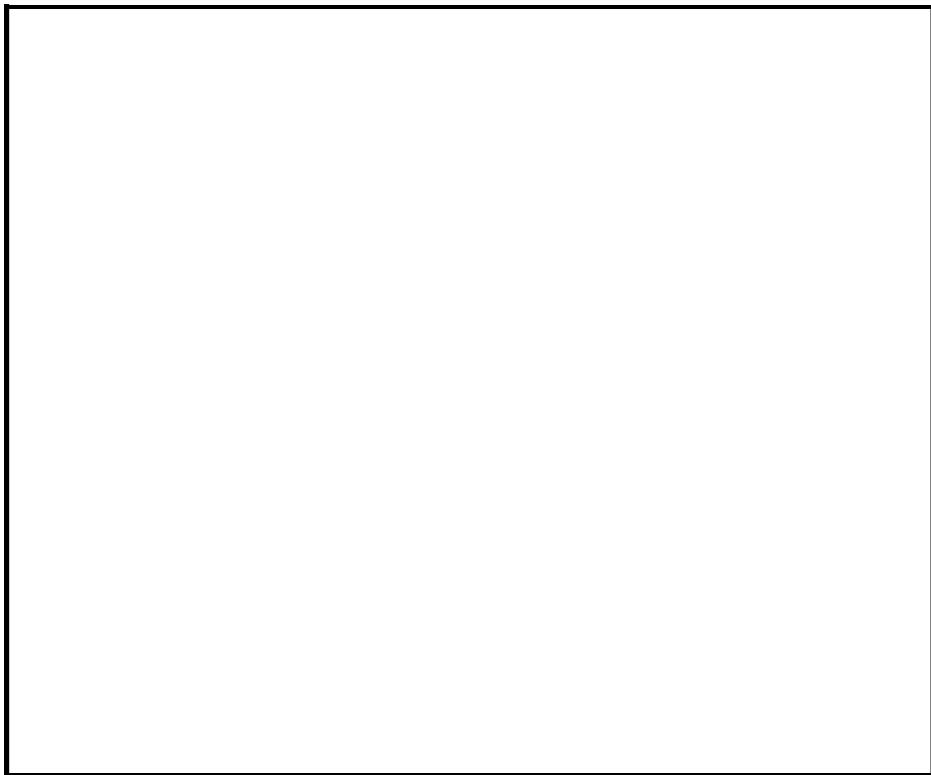
Hacer un programa en Scratch que vaya preguntando y si contestas bien que salga aplausos y un texto "bien" en caso contrario un sonido de fallo y el mensaje "No no"



Solución

Este es el proyecto: <https://scratch.mit.edu/projects/124217131/>

si no tienes el Makey Makey no pasa nada, pulsa con el teclado A y W y a jugar !!!



## Piano

### Propuesta

Hacer un piano humano



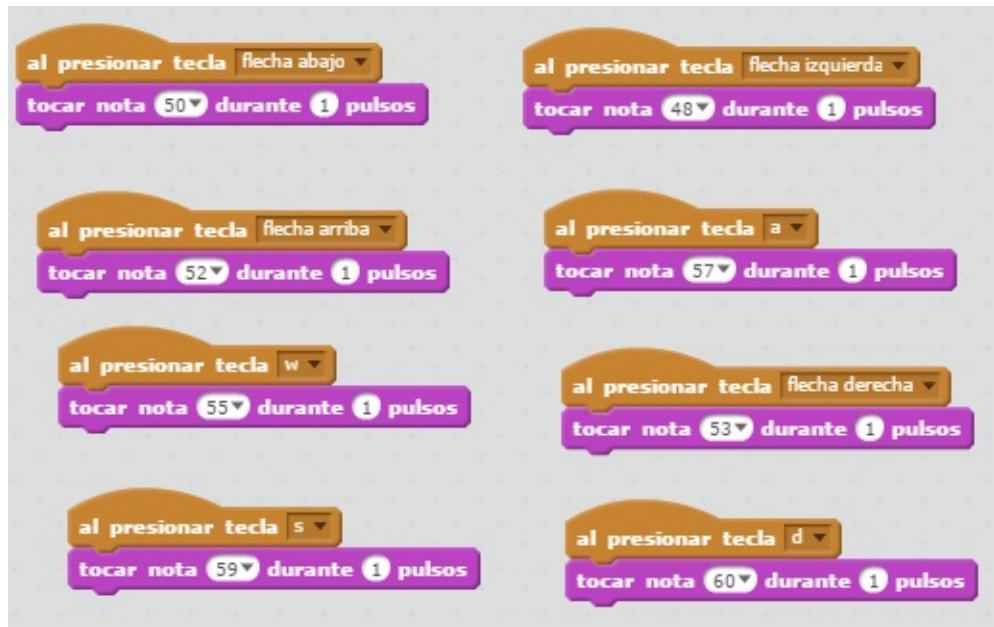
[Video link](#)

Solución

Primero tenemos que definir qué teclas son qué notas, nuestra propuesta es la siguiente:

Tecla	Nota
Flecha izq	48
Flecha abajo	50
Flecha arriba	52
Flecha der	53
w	55
a	57
s	59
d	60

El programa en sí es muy básico:



Puedes probarlo aquí pero utilizando las teclas del teclado y del ratón



La única dificultad es la construcción: Cada tecla del Makey Makey se conecta con un cable, y al otro extremo una persona que hace de esa tecla.

El que toca el piano tiene que estar conectado a la masa del Makey Makey con un cable

Los niños se lo pasan genial:



[Video link](#)

O no tan niños ;)



[Video link](#)

## ALTERNATIVAS

Con bananas, plastilina, gominolas....



Piano

---

[Video link](#)

# El cuerpo humano

## Propuesta

Esta propuesta es de [Programo Ergo Sum](http://www.programoergosum.com/), <http://www.programoergosum.com/> agradecemos su colaboración



[Video link](#)

Solución



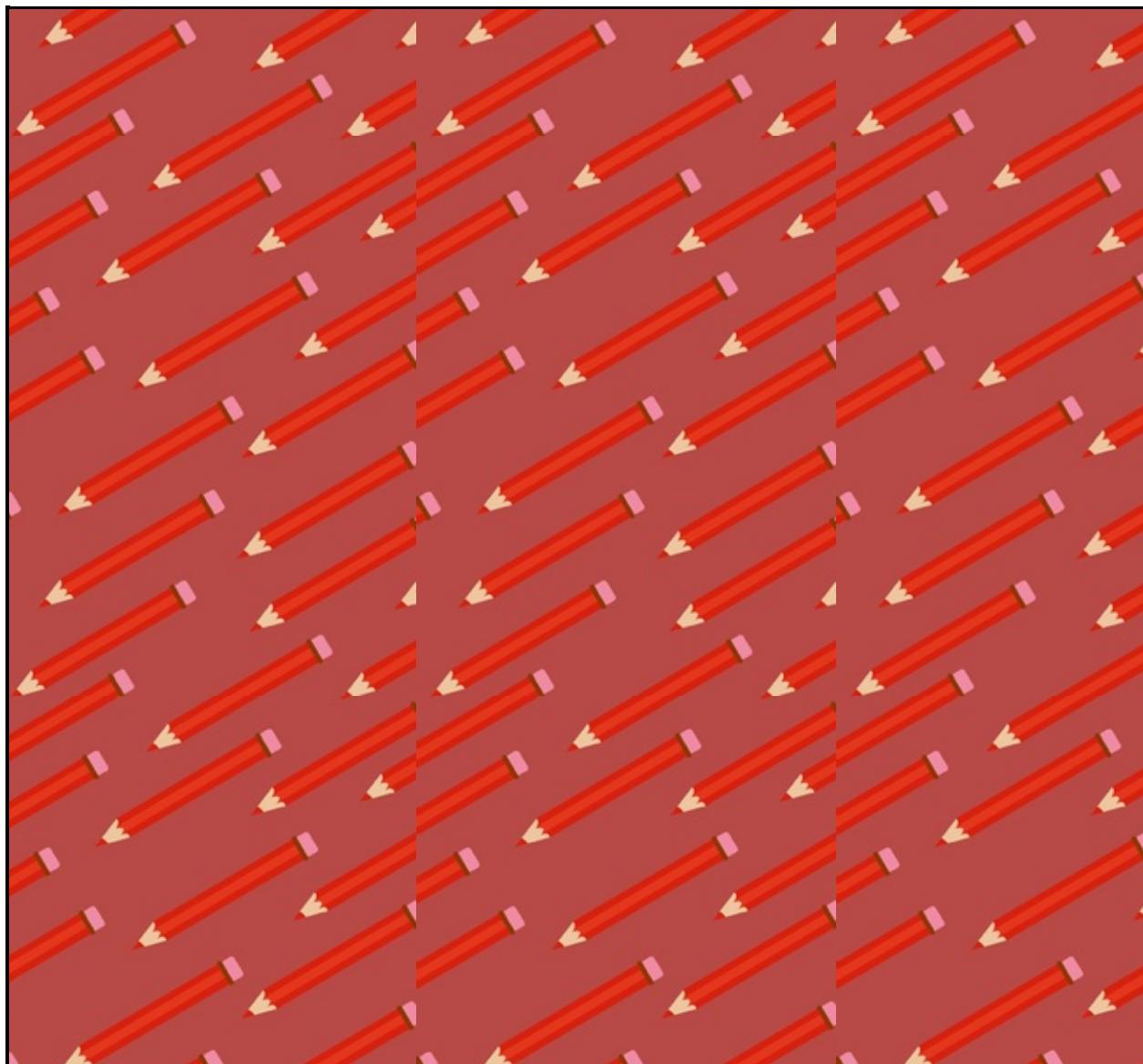
[Video link](#)



[Video link](#)

## Cuelga experiencias

En este muro puedes colgar experiencias interesantes tuyas o que veas por internet <https://padlet.com/CATEDU/makey>



Made with padlet

## ProgramoErgoSum

Esta sección la queremos agradecer al autor de la página <http://www.programoergosum.com/> que nos ha autorizado publicar sus vídeos.



# PROGRAMO ERGO SUM

Contínuamente el autor sube propuestas, recomendamos visitar [su canal de vídeo Youtube](#) y suscribirse para estar al día.

## Vocales

### Propuesta



[Video link](#)

Solución



[Video link](#)



[Video link](#)



[Video link](#)

## Clonar Matar pájaros

### Propuesta

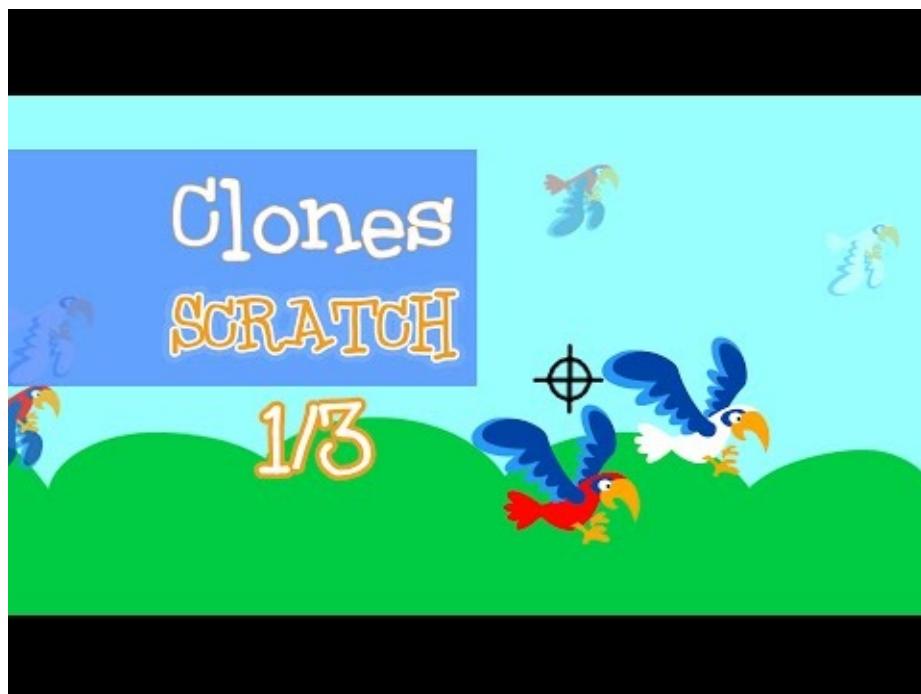
Esta es una buena propuesta de [PROGRAMO ERGO SUM - CLONAR](#)



[Video link](#)

Solución

Preliminares



[Video link](#)

Pájaro azul: clonar, movimiento, punto de mira, vidas ...



[Video link](#)

Ahora el pájaro blanco:



[Video link](#)

## 2 jugadores Fútbol

### Propuesta

Nuestra propuesta es de [Programa Ergo sum - futbol](#)



[Video link](#)

Solución



[Video link](#)



[Video link](#)



[Video link](#)

## Bloques Duck Hunt

### Propuesta

La mejor forma de aprender los bloques es con un ejemplo de [Programa Ergo Sum](#):



Solución

Preliminar



[Video link](#)

Aquí es donde se emplea los bloques



[Video link](#)

y finalizando



[Video link](#)

# Flappy movientos

## Propuesta

La propuesta es de Programo Ergo Sum Flappy



Solución

Fondos



[Video link](#)

movientos



[Video link](#)

puntuación



[Video link](#)

## Pong

### Propuesta



[Video link](#)

### Solución



[Video link](#)



[Video link](#)



[Video link](#)

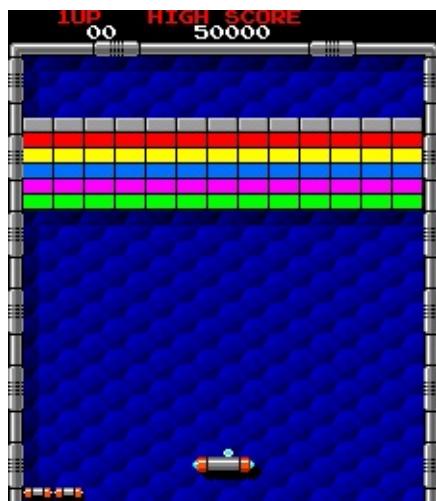


[Video link](#)

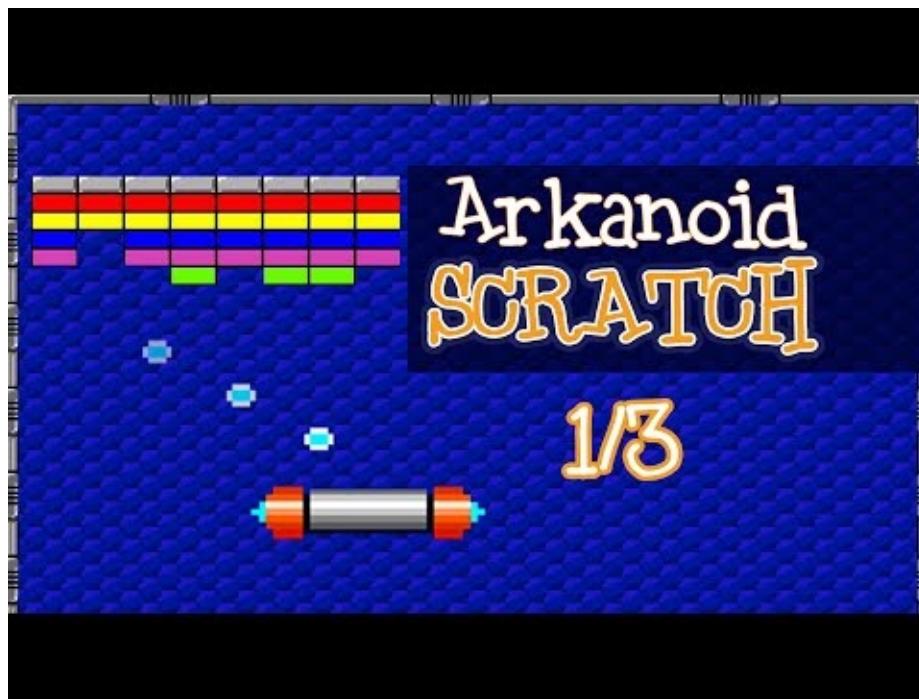
## Rebotes - Arkanoid

### Propuesta

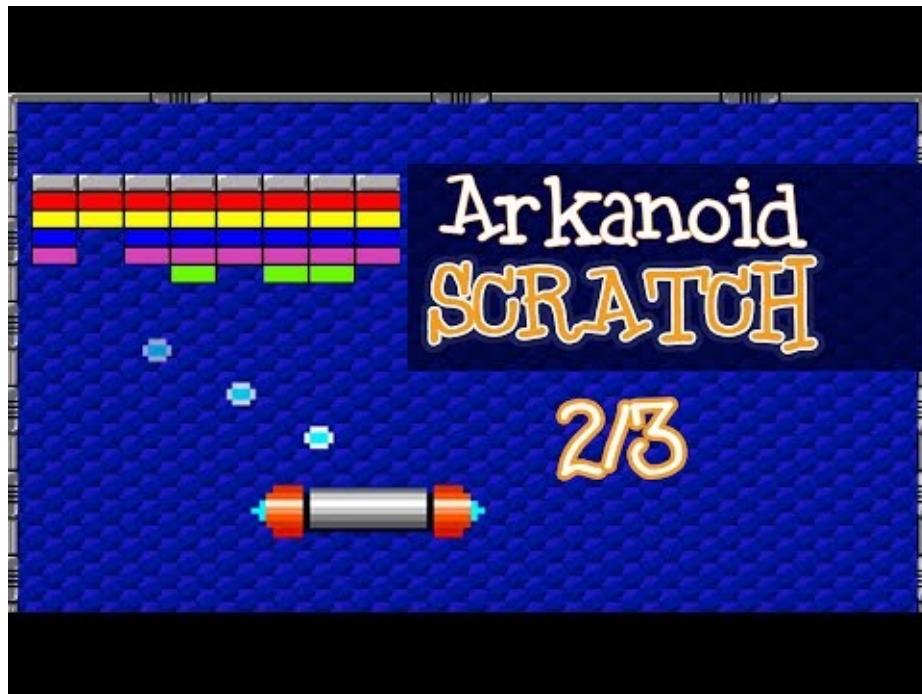
Nuestra propuesta es de [Programo Ergo Sum -arkanoid](#)



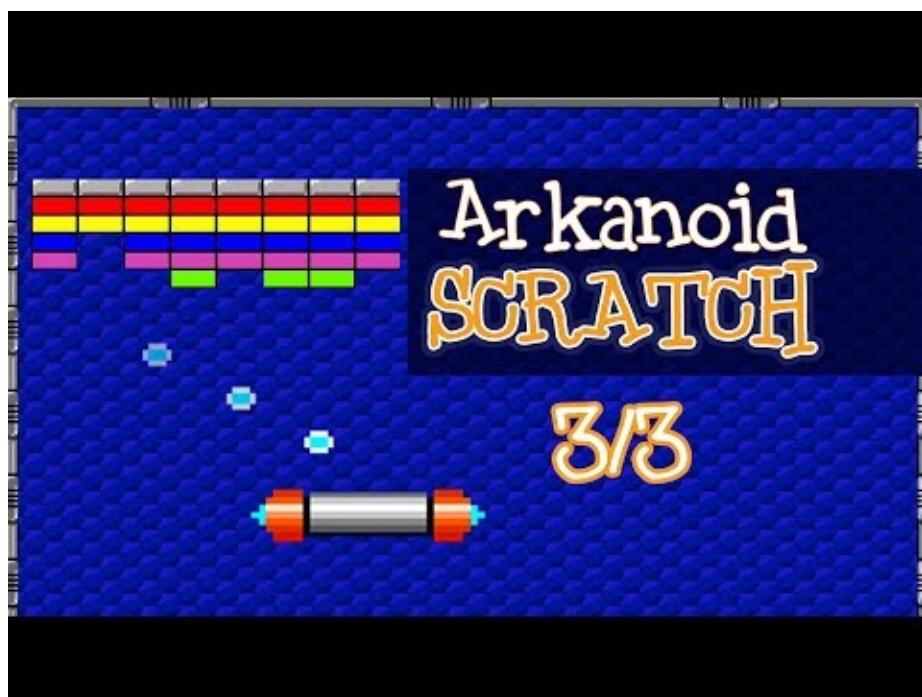
Solución



[Video link](#)



[Video link](#)



[Video link](#)

# Reinventar

Buscamos un programa que esté relacionado con el movimiento del sprit, por ejemplo:

The screenshot shows the Scratch website interface. At the top, there's a navigation bar with links: 'Crear', 'Explorar', 'Comentar', 'Acerca de Scratch', 'Ayuda', and a search bar containing the text 'mover gato'. A yellow oval highlights the search bar. To the right of the search bar are icons for messaging, sharing, and account management, along with the 'catedu' logo. Below the navigation bar is a section titled '¿Qué sucede?' which says 'This is where you will see updates from Scratchers you follow'. To the right is a 'Noticias de Scratch' sidebar with a 'Wiki Wednesday' section. A large, empty workspace area is visible in the center.

Nos gusta pero vemos que se podría mejorar:

The screenshot shows the Scratch script editor. A specific script is highlighted with a red box. The script starts with a 'when green flag is clicked' hat block, followed by a 'say [Type your message here]' block, a 'ask [Question] and wait' block, and two 'say [Type your message here]' blocks. A red callout box points to the first 'say' block with the text 'Se podría mejorar!'. In the top right corner of the editor, there's a yellow button labeled 'Reinventar' with a red border, and a red arrow points from the 'Se podría mejorar!' callout towards this button.

Mejoramos el código, añadiendo una variable "frase\_bienvenida" que recoja la respuesta y el texto que queremos que aparezca



Le damos a compartir y nos sale un mensaje de bienvenida y la opción de describir nuestra mejora:

The screenshot shows a Scratch project titled "jose luis duran remix remix" created by "reinventados por catedu". The project has 13 programs and 4 objects. It features a background image of a stream with a crab, a cat, and a frog. The script includes a welcome message and a note about adding a variable for the user's name.

**Instrucciones:**  
atrapa comida para el gatito, pero cuidado con lo que atrapas

**Notas y créditos (añadido por catedu):**  
He añadido una variable "Frase\_bienvenida" que recoja el nombre del usuario X y así el gato contesta "Bienvenido X, soy el gato MU!"

**Basado en:**  
[jose luis duran remix by mrojgar19](#)

**Proyecto original:**  
[jose luis duran by jdurhid20](#)

Add project tags.

Compartido en: 29 Sep 2016 | Modificado: 29 Sep 2016

0 stars | 0 hearts | Estudios | Embebido | 1 views | 3 likes

y ahora es otro proyecto: <https://scratch.mit.edu/projects/123302249>



0

object tags.

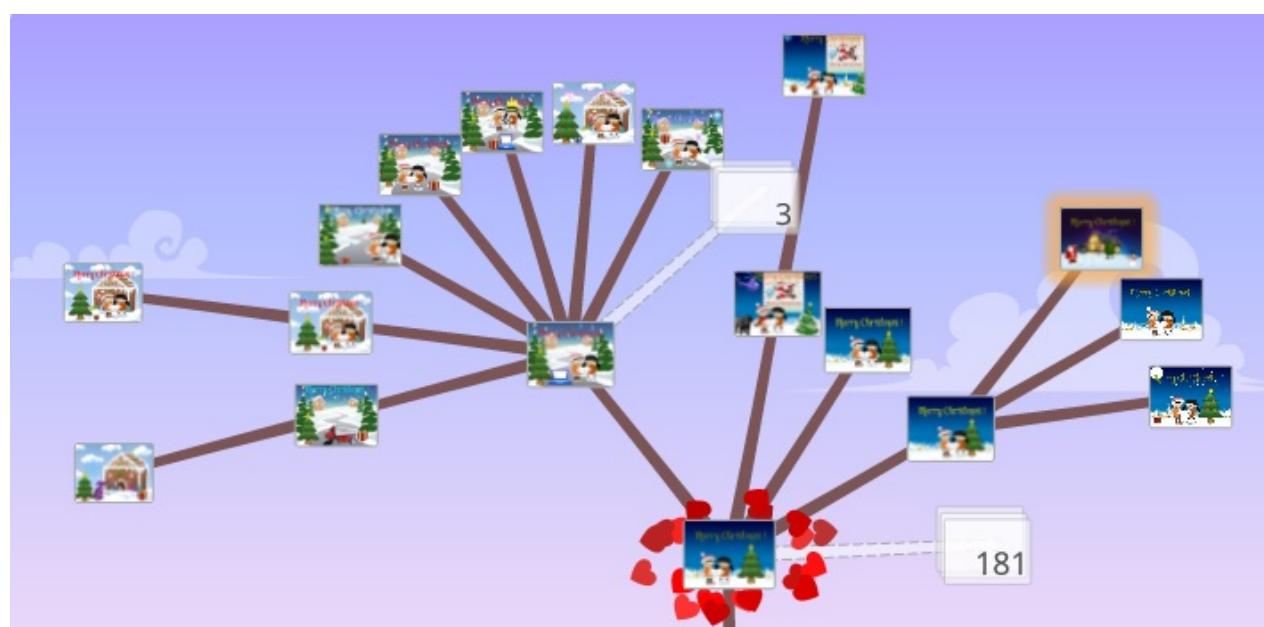
Modificado: 29 Sep 2016

1

3

Si pinchamos en el árbol sale [esta página](#) y aparece 3 reinversiones (a fecha 29/9/16) el primero es el original, uno segundo (del mismo autor que lo mejoró) y esta última de Catedu

Hay proyectos donde las reinversiones son muchas y salen muchas versiones de la misma, por ejemplo [en este proyecto](#) de tarjetas navideñas [el árbol de reinversiones](#)





## ¿Por qué reinventar?

Recordemos que [Scratch es código libre](#),) por lo tanto es [software libre](#) de ser copiado, estudiado o modificado. Esto no debe de confundirse con [OpenSource](#) o código abierto, con licencia abierta para que técnicamente se mejore, o [FreeWare](#), es decir gratis.

Aunque técnicamente no se esté obligado a mejorar el código como en OpenSource, sí que se obliga a acceder al código fuente para poder copiarlo y por lo tanto tenemos libertad de reinventar, podemos obtener una versión diferente que no necesariamente nos obligue a una mejora, somos libres de hacer con otros programas de otros usuarios lo que queramos mientras respetemos la licencia libre (acceso al código principalmente)

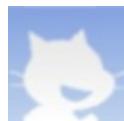
Se puede aprovechar la actividad de reinventar en el aula para introducir los conceptos de mejora de código comunitarios:

- Los 4 principios filosóficos del software libre. ([ver](#))
- Los 10 principios técnicos del software abierto ([ver](#))
- El software gratis no es libre o abierto y viceversa
- La mejora de la comunidad gracias al acceso de las licencias [GNU o GPL es decir Licencia Pública](#)
- La historia de [GNU/Linux](#)
- Desarrollo de programas libres frente a los privativos



	<b>GNU</b>	<b>PROPIETARIO</b>
SISTEMA OPERATIVO	 	
NAVEGADOR	 Mozilla Firefox	 Google Chrome
SUITE OFIMÁTICA	 Libre Office	
REPRODUCTOR DE VIDEO	 VLC media player	 Reproductor Windows media player
REPRODUCTOR DE AUDIO	 Songbird	 iTunes
EDITOR DE IMÁGENES	 Gimp	 Photoshop
EDITOR DE VIDEOS	 Avidemux	 Adobe premiere pro

## Conviertete en un Scratcher



Al cabo de un tiempo Scratch te preguntará si quieres ser un **Scratcher** ¿qué implica? pues básicamente que respetarás las normas básicas de convivencia un foro de este tipo, actualmente se fija en los siguientes aspectos:

- Limitaciones en tus **comentarios**: Tus críticas tienen que ser constructivas, no negativas y respetando las opiniones, trabajos y comentarios de los demás
- Limitaciones en **tiempo de respuesta** en los debates (dirigidos a evitar spam)
- Limitaciones en el uso de **enlaces externos**, imágenes, almacenamientos en la nube, etc... evidentemente sí que puedes usar enlaces a otros proyectos Scratch

El Scratcher, por supuesto tiene que estar registrado y aunque no es necesario, lo normal es que comparta proyectos, haga comentarios de otros trabajos, reinvente... es decir participar en la comunidad Scratch.

Aquí se puede trabajar los **valores de las comunidades virtuales**, la importancia de los comentarios, foros, debates ... sin perder la vista el **constructivismo, colaboración** ... no la simple palabrería. La cooperación, no sólo técnicamente, sino también se puede ser animar, felicitar ...

[https://wiki.scratch.mit.edu/wiki/New\\_Scratcher\\_Status](https://wiki.scratch.mit.edu/wiki/New_Scratcher_Status)

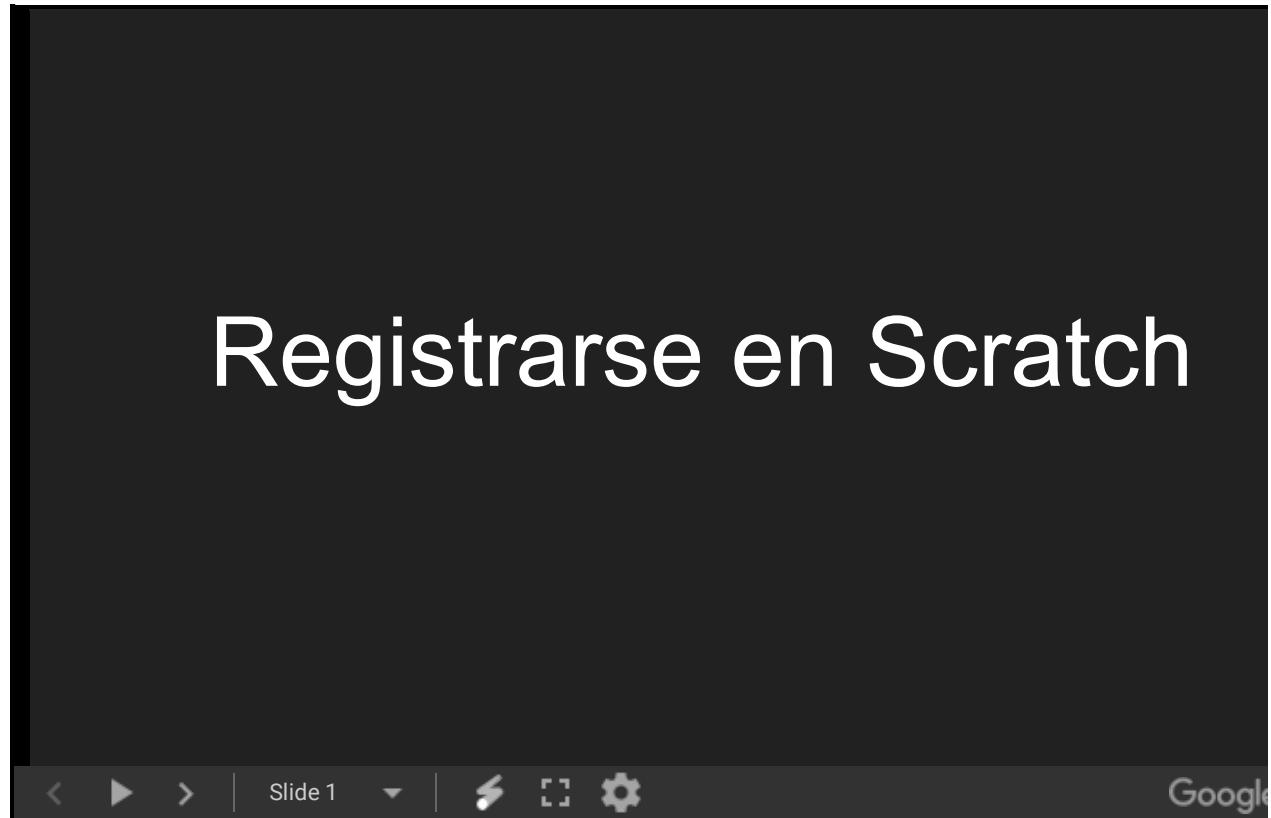
## Scratch para docentes



## Abrir cuenta como profesor

Es distinto registrarnos como profesor que como alumno, incluso un usuario ya creado no puede convertirse en profesor, hay que empezar desde el principio.

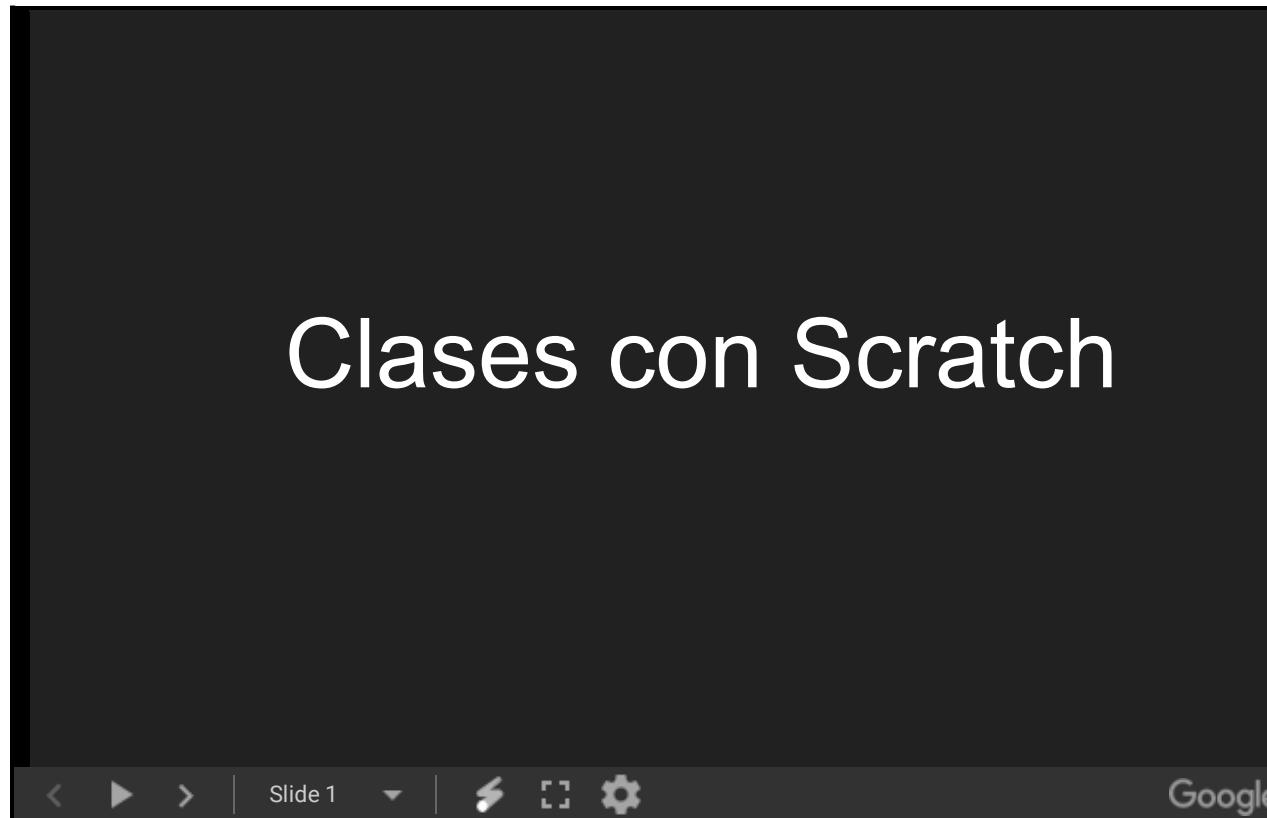
Aquí tienes un tutorial [en formato flash](#) o [en pdf](#) si no lo ves bien:



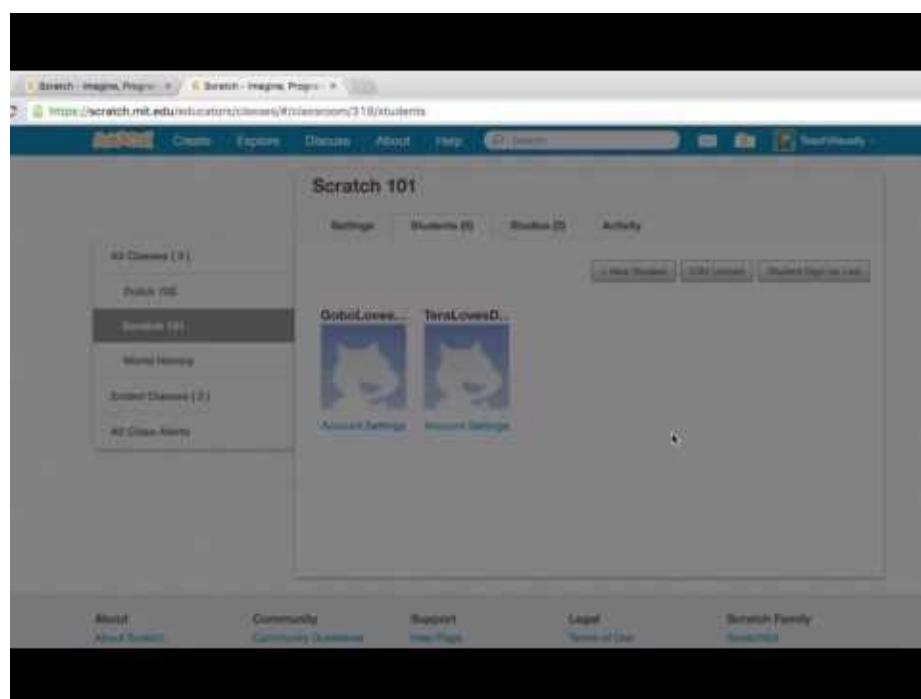
## Clase con Scratch

Pasado 24 horas, puedes averiguar si han aceptado tu cuenta de profesor ([ver tutorial anterior](#)) y empezar a trabajar.

Para ver un poco las posibilidades, tienes este [tutorial en FLASH](#) o en [PDF](#) como siempre de elaboración propia:



También puedes verlo en este vídeo de Youtube de Scratch pero en inglés:



[Video link](#)



## Evaluar

Para evaluar un proyecto con Scratch podemos hacer una rúbrica más o menos elaborada como en [este proyecto de música](#), pero implica un esfuerzo considerable. Proponemos:

### **Dr Scratch <http://www.drscratch.org/>**

Es una forma objetiva, rápido aunque sólo evalúa el aspecto informático de programación, no los conceptos, la imaginación, los sprits y escenarios creados...

Podemos ingresar el proyecto mediante la URL (perfecto si utilizamos Scratch online) o por archivo (si utilizamos Scratch offline)

Evaluá del 0-3 estos ítems: Paralelismo, Pensamiento lógico, Control de flujo, Interactividad con el usuario, Representación de la información, Abstracción , Sincronización

En total del 0 al 21, incluso da una realimentación de mejoras

Si nos registramos, graba un historial de logros.

### **Un ejemplo:**

Si evaluamos este proyecto <https://scratch.mit.edu/projects/123355627/>



Si realizan la evaluación en [Dr Scratch](#): pues nos sale ... oh cielos !!! que malo que soy !!! por favor que esto no se publique !!!



Puntuación: 3/21

[Tweet](#)

El nivel de tu proyecto es...  
**¡BÁSICO!**

Estás al principio de una gran aventura...  
¡Continúa así!

AYUDA DR. SCRATCH(VERSIÓN BETA)

Mejora tu nivel	Nivel
💡 Paralelismo	0/3
💡 Pensamiento lógico	0/3
💡 Control de flujo	1/3

Y encima da opción a descargar un certificado en PDF ¡qué vergüenza !



## Consejos

### Para empezar Scratch en clase

- Fichas de Scratch [descarga de la fuente](#)
- Fichas de Scratch [Aprende como un niño](#)
- Los 12 retos cortos de menor a mayor profundidad [descarga](#) (doc - 337.5) fuente
- Para los más pequeños, con tableta y la app Scratchjr [hacer estos retos](#)
- Ideas para principiantes en [https://scratch.mit.edu/starter\\_projects/](https://scratch.mit.edu/starter_projects/)
- Paso a paso en Scratch online [https://scratch.mit.edu/projects/editor/?tip\\_bar=getStarted](https://scratch.mit.edu/projects/editor/?tip_bar=getStarted)

### A la hora de proponer un videojuego en clase

Tal y como vistes en la [presentación del módulo 1](#) :

Tiene que ser ejemplos que funcionen, adaptados a su capacidad. ¿Como se consigue esto?

- Entra en Scrach on line
- Mira ejemplos que funcionen
- Mira por dentro la programación si es adecuado a su capacidad
- A ellos propon la idea, de tal manera que sea difícil encontrar su solución exacta en Scratch

Fijar los objetivos ¿de que va?¿por qué es divertido?

Usar escenarios, diseñarlos, no entretenerte mucho en esto pues se escapa del pensamiento computacional.

Escribir la lógica en un papel, detectar inconsistencias, pedir ayuda.

### Enlaces

Curso en EducaMadrid

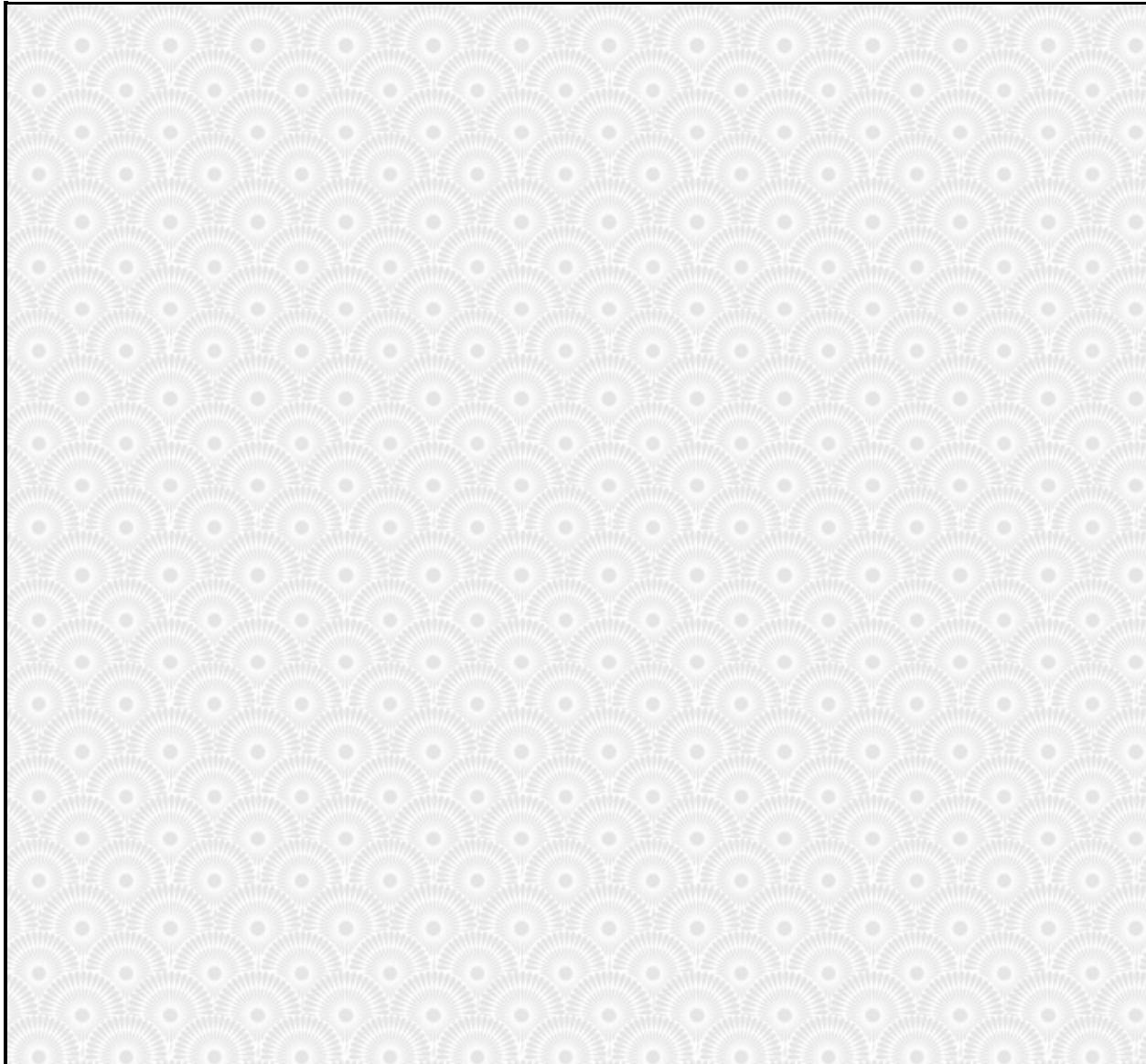
- [Prácticas básicas con Scratch 1.](#)
- [Prácticas básicas con Scratch 2.](#)
- [Prácticas básicas con Scratch 4.](#)
- [Prácticas básicas con Scratch 5.](#)

Guías:

- En [Eduteca](#)
- [Juan Carlos López 01/05/13](#)
- [Ceip Cella Teruel](#)
- [Gobierno de Canarias, con enlaces a diversas áreas Matemáticas, Lenguaje, Sociales](#)
- [Cefire de Elda](#)

## Retos

Aquí tienes un muro donde puedes encontrar todos aquellos ejemplos que veas interesantes para realizar en Scratch



Made with nodlet

2017 por [CATEDU](#) (Daniel Pons Betrián y Javier Quintana Peiró).

Cualquier observación o detección de error por favor aquí [soporte.catedu.es](mailto:soporte.catedu.es)

Los contenidos se distribuye bajo licencia Creative Commons tipo BY-NC-SA.



# GOBIERNO DE ARAGÓN

Departamento de Educación,  
Cultura y Deporte

