

3. Entradas de Echidna

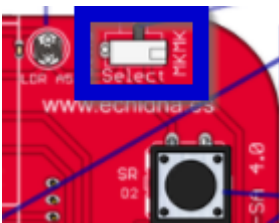
- [Otra vez acuérdate](#)
- [MONTAJE 4 ENCENDIDOAPAGADO](#)
- [MONTAJE4-ENCENDIDOAPAGADO-AVANZADO](#)
- [MONTAJE 5 LDR](#)
- [MONTAJE 5 SEMAFORO SONIDO](#)
- [MONTAJE 6 Piano luminoso](#)
- [MONTAJE 7 Telesketch](#)
- [Mapeo](#)
- [MONTAJE 8 Comebichos](#)
- [MONTAJE 9 SPRITEVOLADOR](#)
- [MONTAJE 10 MATA-ALIENS hackeando código](#)

Otra vez acuérdate

Aquí es donde vamos a sacar todo el jugo a esta Shield.

Nota: Si has hecho el anterior programa Grabar a Arduino, recuerda dejarlo preparado para que haga caso a mBlock: Conectar-Actualizar Firmware.

Nota: Acuérdate en toda esta sección de poner la Echidna en modo Sensor



Todos los programas de este curso se encuentran en este repositorio:

<https://github.com/JavierQuintana/Echidna>

MONTAJE 4

ENCENDIDO APAGADO

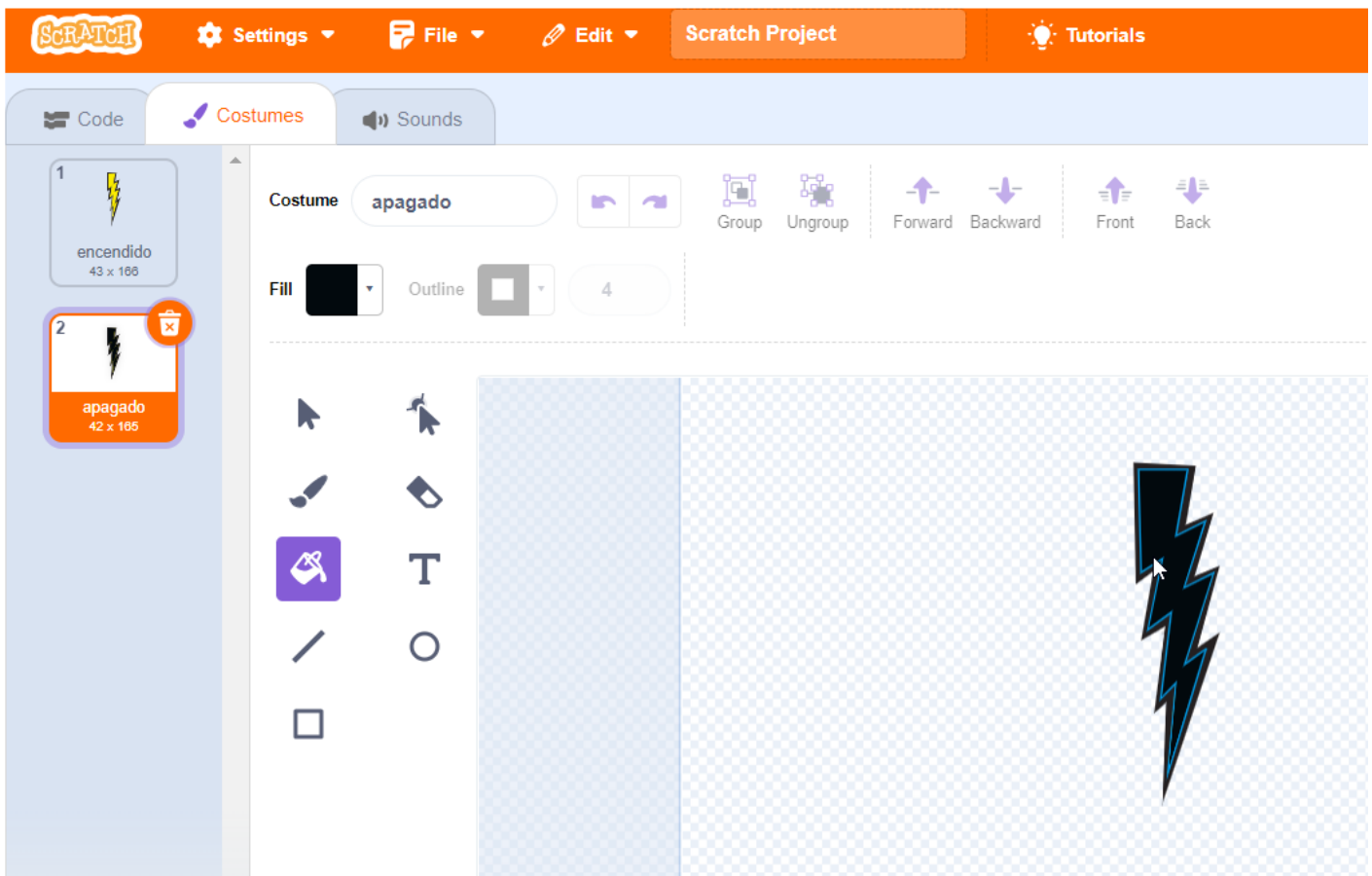
El anterior programa ya es un ejemplo de uso de estos botones digitales que están conectados a los pines digitales 2 y 3 de Arduino. Ten en cuenta que *sólo* pueden leer niveles lógicos.

MONTAJE4-ENCENDIDO APAGADO

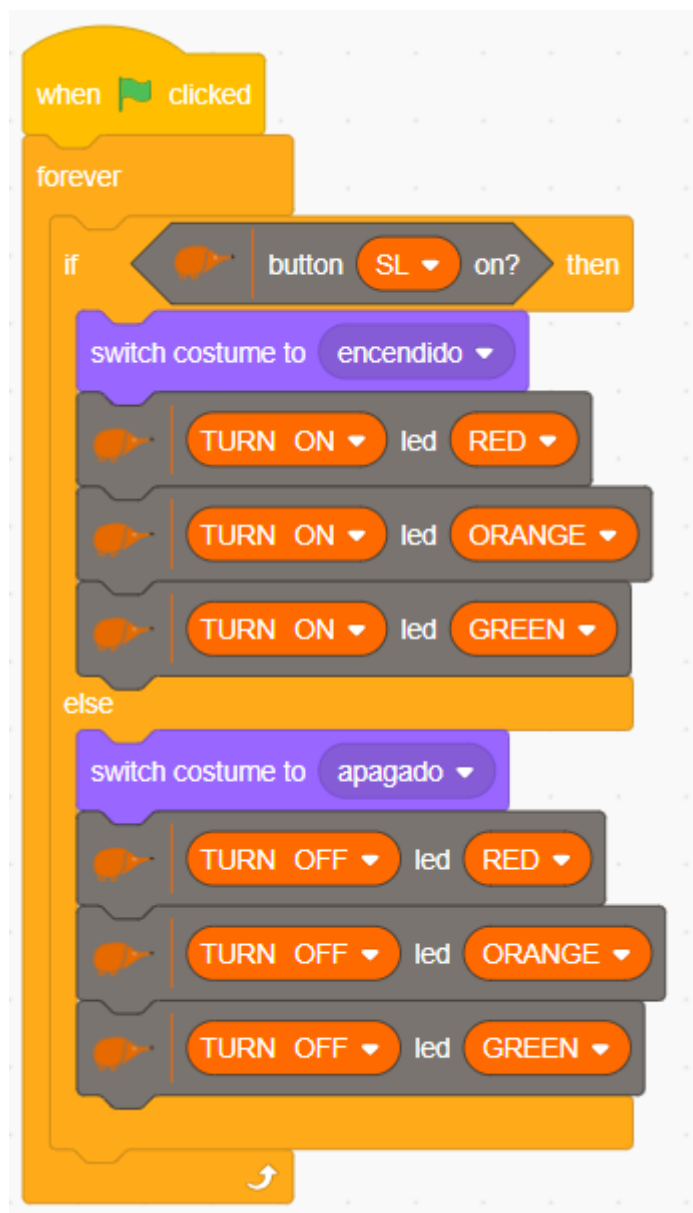
Al pulsar el botón D2 se enciende los 3 leds del Echidna y al soltarlo se apagan. El programa también tiene que tener un efecto en un sprite

Solución con EchidnaScratch

Cambiamos el sprite del gato por otro más acorde y cambiamos el disfraz, no lo desarrollamos aquí pues vimos cómo se hace con el MONTAJE1



El código :

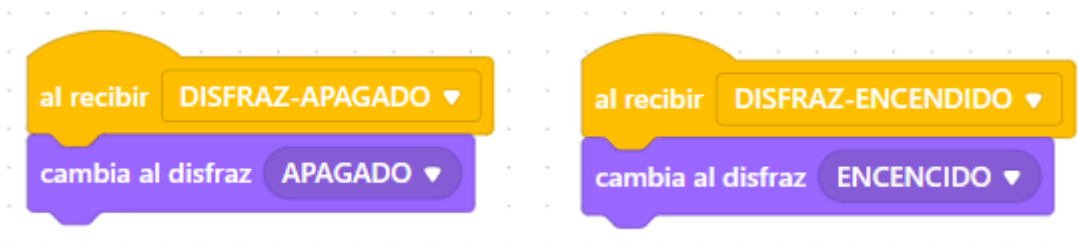


https://www.youtube.com/embed/9pn_a5v6PA4

El fichero lo tienes en el Github : <https://github.com/JavierQuintana/Echidna>

Solución con mBlock

Elegimos un objeto que cambie el disfraz, y le ponemos el código



Y al arduino:



El programa lo tienes aquí <https://planet.mblock.cc/project/3228733>

MONTAJE4- ENCENDIDOAPAGADO- AVANZADO

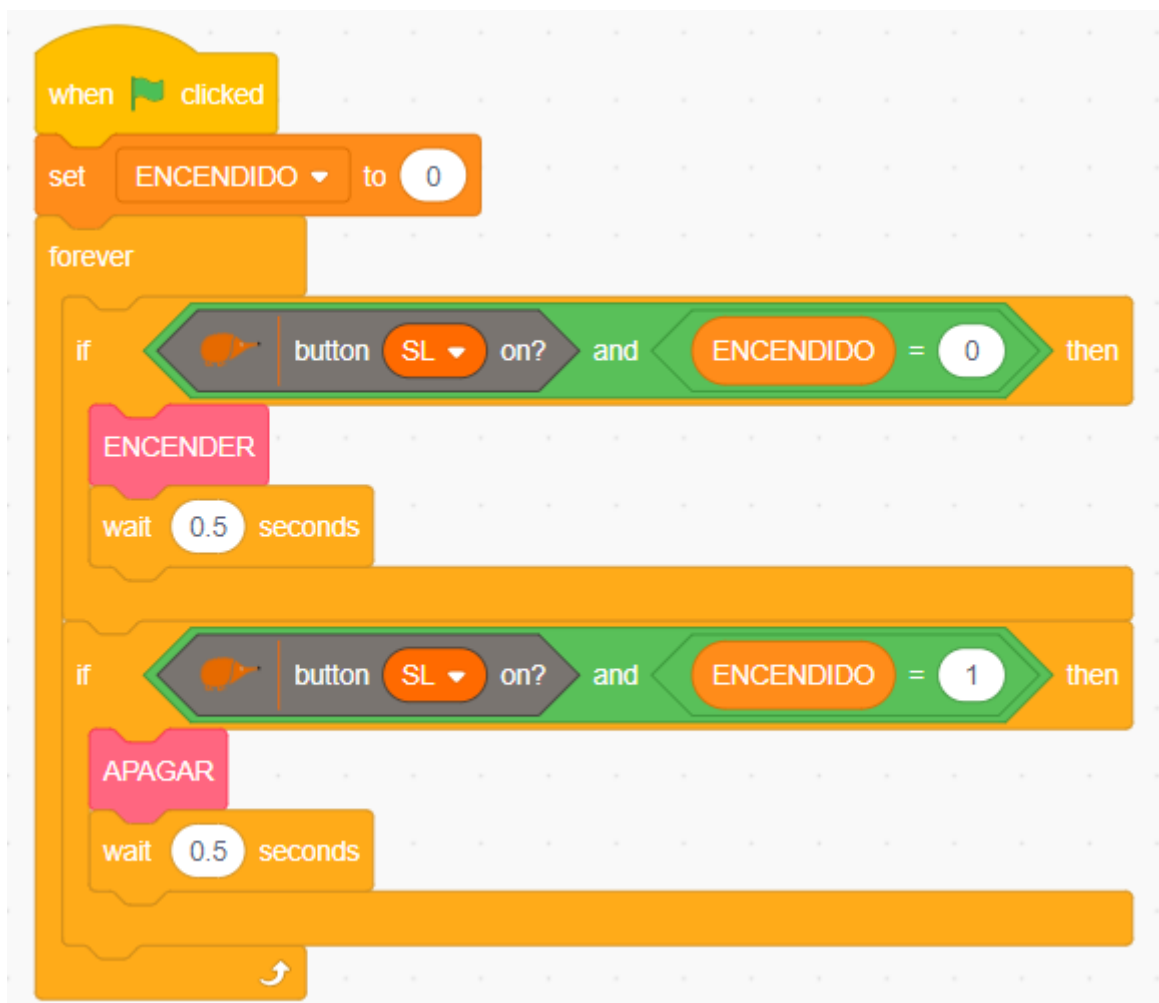
MONTAJE4-ENCENDIDOAPAGADO-AVANZADO

Al pulsar D2 tiene que encenderse los leds, y **sólo se apagan** si se vuelve a pulsar D2. O sea, utilizar sólo un pulsador para encender y apagar los leds.

Solución EchidnaShield

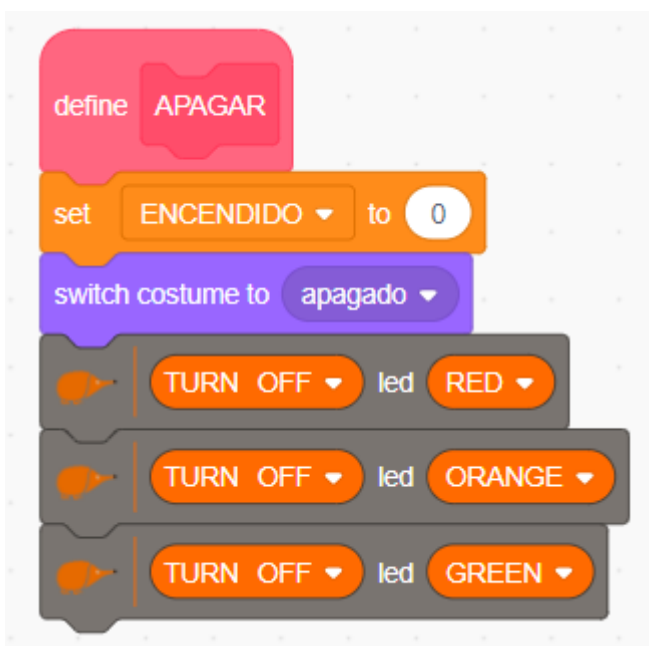
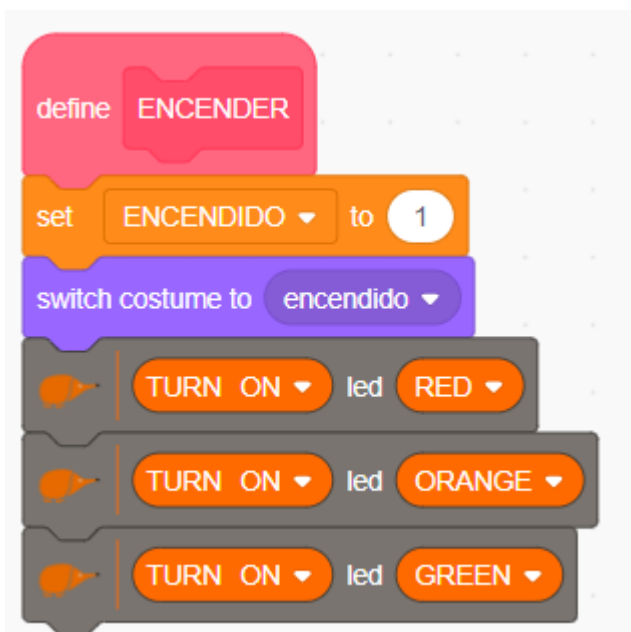
En este caso creamos una variable **ENCENDIDO** que registra si están los leds y el sprite encendidos o no

El código es



Para evitar que el programa sea engorroso, se ha creado los bloques ENCENDER y APAGAR, pero perfectamente se puede meter estas instrucciones dentro del algoritmo principal y pasar de bloques.

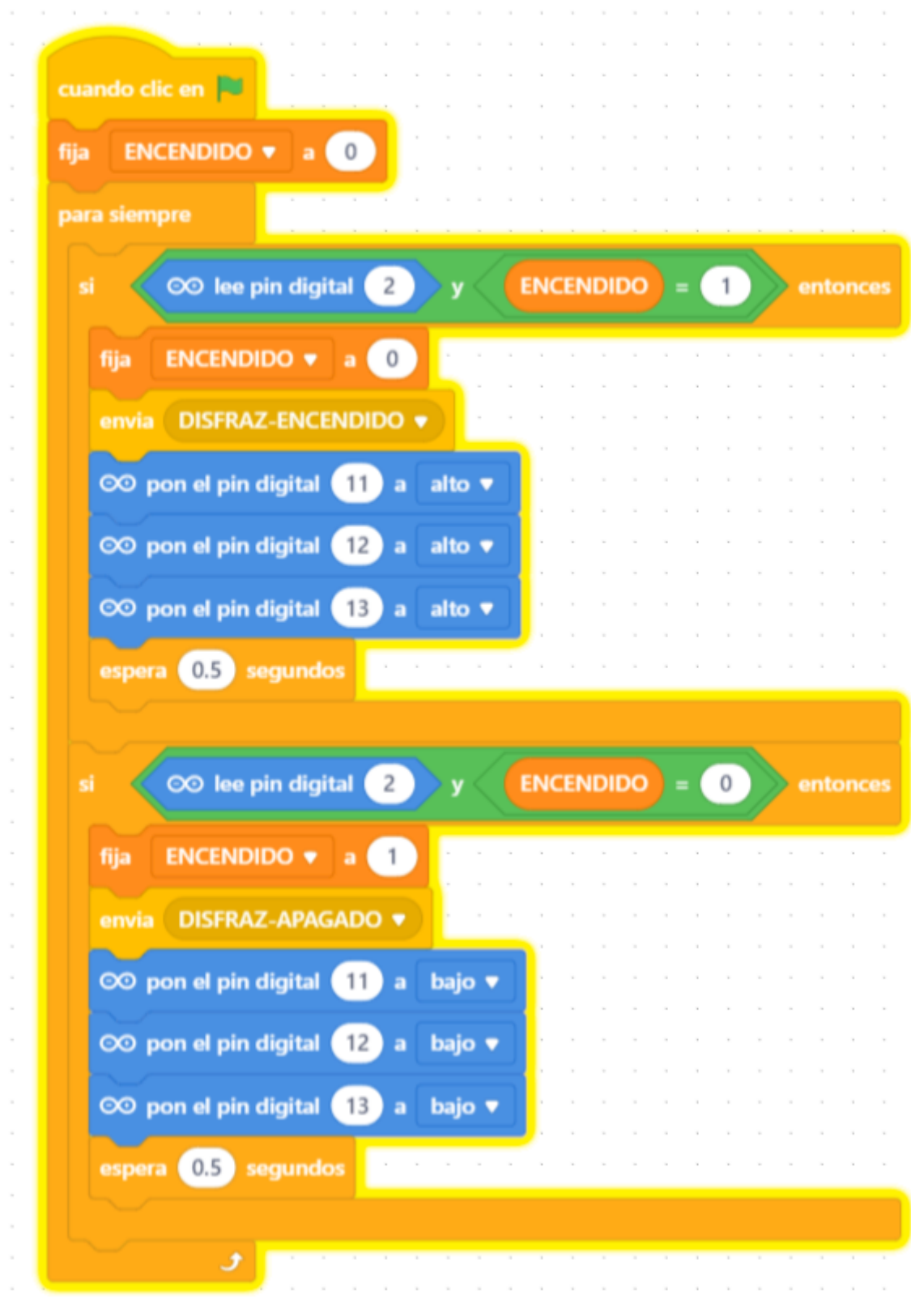
FIJATE que la variable ENCENDIDO se cambia a 1 o a 0 para que refleje el estado en el que ha quedado el sistema



<https://www.youtube.com/embed/ZH-nsYdAazY>

El programa lo tienes en : <https://github.com/JavierQuintana/Echidna>

Solución con mBlock



El objeto que cambia el disfraz es igual que en RETO1

El programa lo tienes aquí <https://planet.mblock.cc/project/3228741>



UNA CURIOSIDAD Fíjate como hay un retraso de 0.5 segundos para que "te de tiempo de apartar el dedo del pulsador" en caso contrario, volvería al otro estado. Si no sabes lo que quiero decir, quita la instrucciones "Espera 0.5 segundos" y verás qué pasa.

UN POCO DE PARÉNTESIS TEÓRICO Fíjate en el enunciado del RETO1: "al pulsar el botón D2 se enciende y al soltarlo se apaga" ES UNA MÁQUINA LÓGICA pues el estado de la máquina sólo depende de las entradas (en este caso de un botón): Pulsar la entrada (botón D2) produce una salida concreta (encender leds).

CONTINUAMOS ...

Pero tal y como está redactado, el RETO2 tiene que memorizar el estado anterior, no es trivial el enunciado "Al pulsar D2 tiene que encenderse los leds, y sólo se apagaran si se vuelve a pulsar D2." ES UNA MÁQUINA SECUENCIAL pues el estado de la máquina depende de las entradas y de lo que ha pasado antes. Pulsar la entrada (botón D2) NO produce una salida concreta (depende si estaba apagado o encendido anteriormente).

No pasa nada si no lo entiendes del todo, es teoría.

La programación se complica **necesitamos añadir una variable que recuerde lo que ha pasado antes** la vamos a llamar `_encendido_` que recordará si está encendido los leds o no:

Muchos de nuestros aparatos electrónicos se encienden y se apagan con el mismo botón, así que a partir de ahora aprecia que su funcionamiento no es trivial.

MONTAJE 5 LDR

¿Qué es un LDR?

Vamos a la Wikipedia:



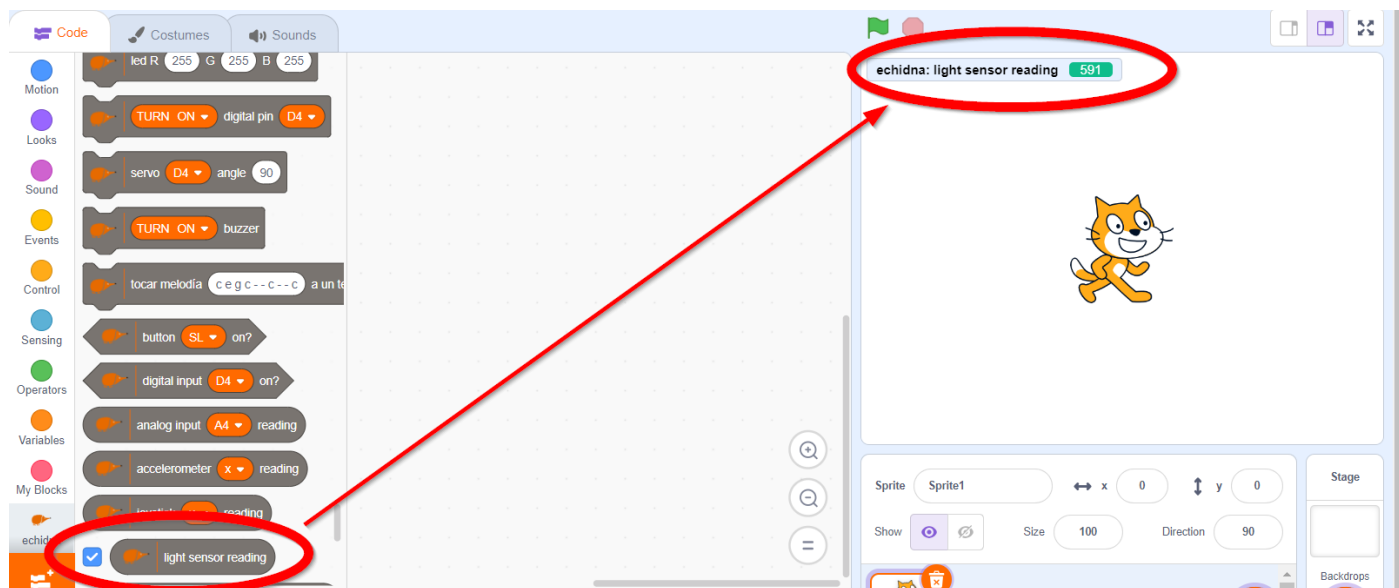
“ [Fotorresistor](#) Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado fotorresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuyas siglas, LDR, se originan de su nombre en inglés light-dependent resistor. Su cuerpo está formado por una célula fotorreceptora y dos patillas. ^W

[More at Wikipedia \(ES\)](#)

Lo verás en el Echidna arriba un poco a la derecha, y está conectado a la entrada analógica del Arduino A5 y según <https://echidna.es/> los valores van desde 20 en ausencia de luz, hasta 1.000 con mucha luz.

Comprobar los límites

No os creáis al pie de la letra los límites oficiales del LDR! Los componentes electrónicos no son ideales, cada uno es particular, probar esos límites en vuestro Echidna



¿Qué vamos a hacer?

Jugar un poco con vuestras condiciones de luz, y definir unos valores de oscuridad, poca luz, media luz y mucha luz, en mi caso mis valores son

- Oscuridad menor 200
- Poca luz entre 200 y 500
- Media luz entre 500 y 800
- Mucha luz mayor 800

Con estos valores vamos a realizar un programa que

- Oscuridad: Ningún led encendido
- Poca luz Luz roja
- Media luz Luz naranja
- Mucha luz Luz verde

MONTAJE 5 Semáforo luminoso

RETO Vamos a reutilizar esfuerzos: reutilizar el semáforo visto pero vamos a hacer que se iluminen los colores según la luz:

Semáforo	Luz
Todo apagado	Mucha oscuridad
Rojo	Oscuridad
Amarillo	Luz normal

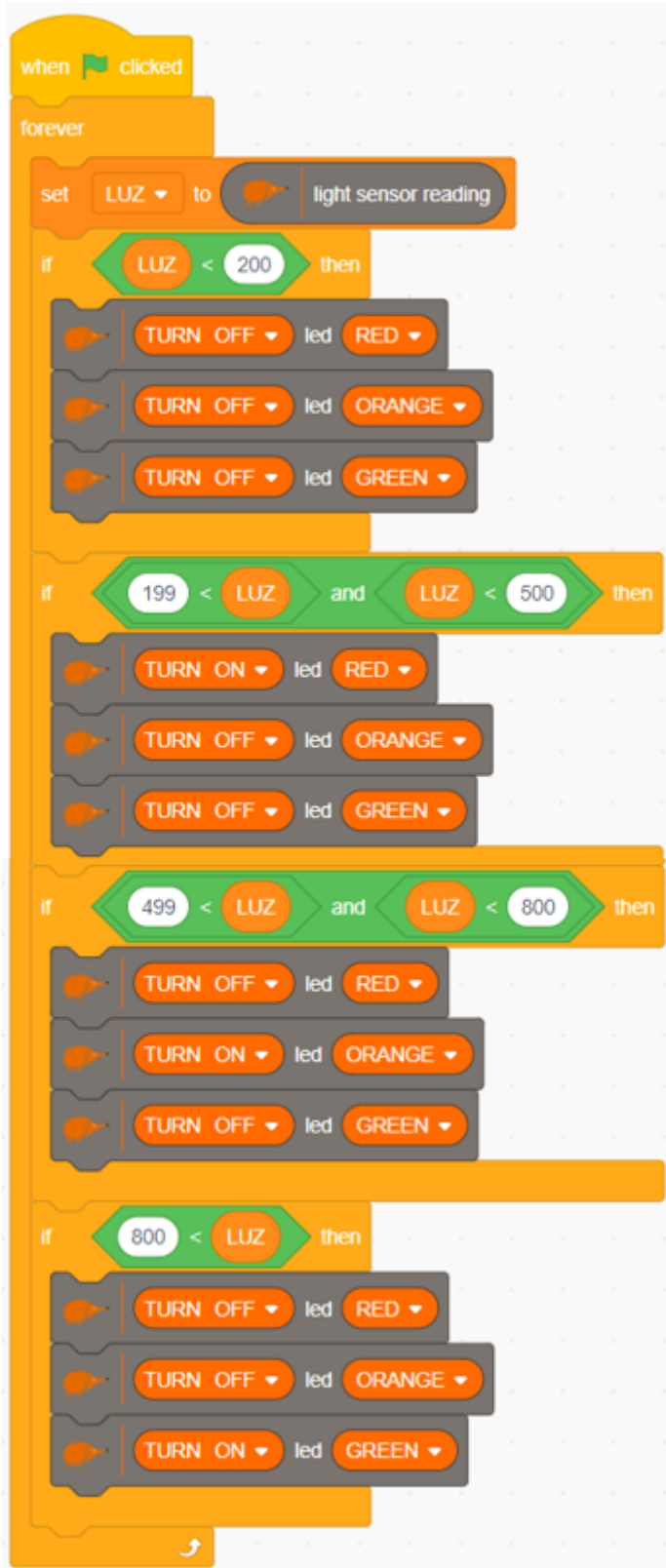


Semáforo	Luz
Verde	Mucha luz

Según nuestros valores experimentales, (puedes poner otros según las condiciones de luz de tu aula)

Semáforo	Luz	Límite inferior	Límite superior
Todo apagado	Mucha oscuridad	---	199
Rojo	Oscuridad	200	499
Amarillo	Luz normal	500	799
Verde	Mucha luz	800	---

Solución con EchidnaScratch

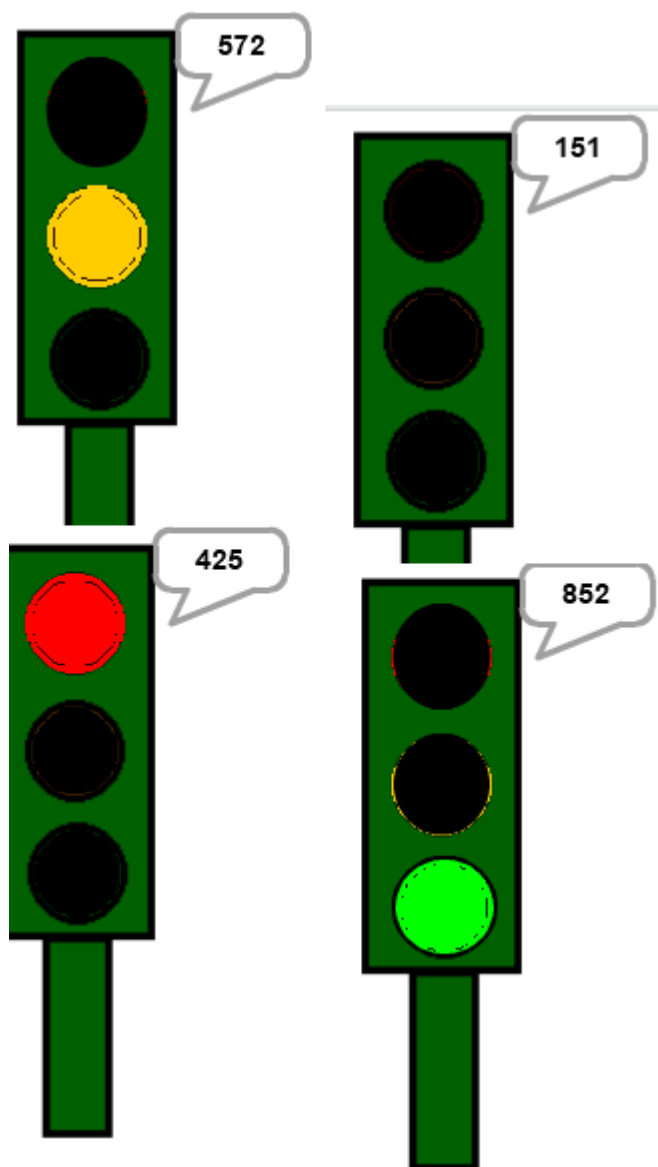


<https://www.youtube.com/embed/5JJY39A2Fm8>

Lo puedes encontrar en en este repositorio: <https://github.com/JavierQuintana/Echidna>

Reto

Haz un sprite tipo semáforo que también se enciendan las luces igual que el Echidna real, tal y como hicimos en MONTAJE1 SEMAFORO y encima que diga los valores



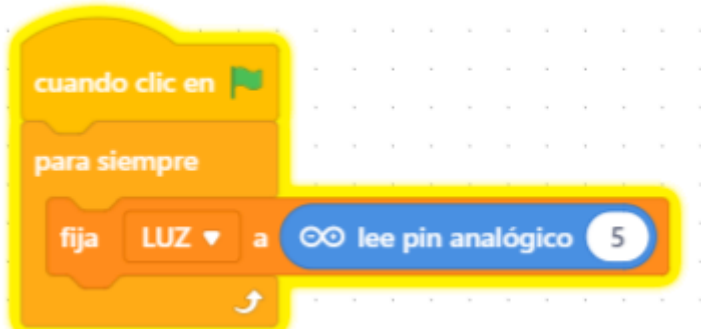
El resultado lo puedes ver en [este vídeo](#):

https://www.youtube.com/embed/MX558VKV_pE



Solución con mBlock

Con mBlock al no tener instrucciones específicas para Echidna, para leer en directo los valores del LDR, hay que utilizar nuestro mBlock y que nuestro simpático oso panda nos diga esos valores, con este sencillo programa en el decide dispositivo **Arduino**:



DONDE **LUZ** ES UNA VARIABLE GLOBAL QUE LO LEEN TODOS LOS OBJETOS luego ahora el panda con este programa puede decirnos cuánta luz hay



El valor que leemos con máxima luz (utilizando una linterna) y el valor de máxima oscuridad (a tapar con el dedo, no te compliques) no llegan a los límites que se marcan oficiales en [Echidna](#) EN TU CASO PUEDEN SER OTROS !! pero parecidos.

El programa lo tienes aquí <https://planet.mblock.cc/project/3228782>

Programa mBlock





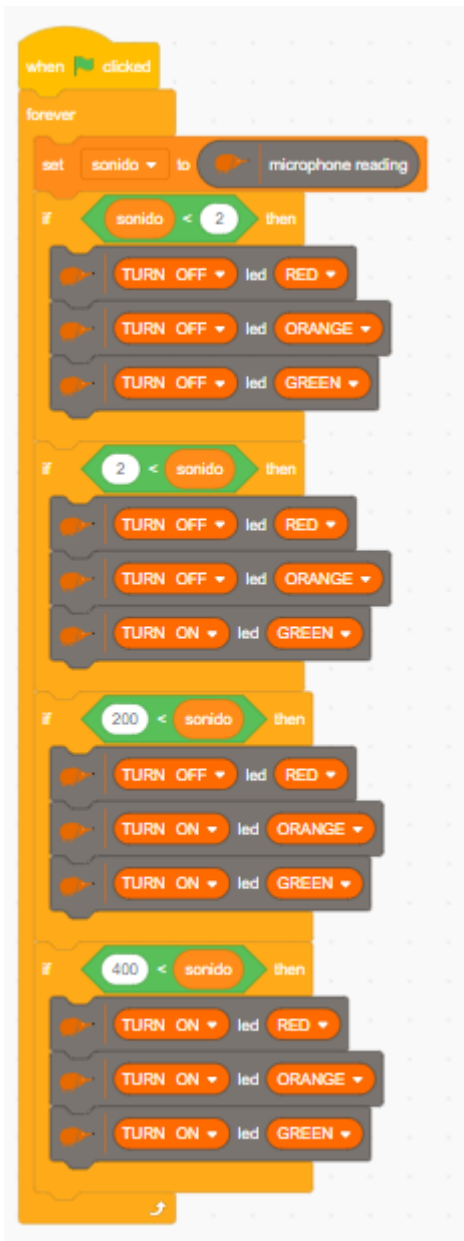
El programa lo tienes aquí <https://planet.mblock.cc/project/3228793>

MONTAJE 5 SEMAFORO SONIDO

No vamos a dejar sin poder experimentar este sensor que mide la intensidad del sonido y como está conectado a la salida analógica A7 sus valores van desde el 0 al 1024

Vamos a hacer un semáforo que mida la intensidad del sonido, verde si es poco, y sube hasta rojo cuando es elevado

El programa en EchidnaScratch es :



Hemos simulado el sonido soplando en el micro :

<https://www.youtube.com/embed/bi17ruyNxSg>

¿Te gustaría este semáforo pero que se ejecute **de manera independiente** (sin necesidad de ordenador), para usarlo en el aula, comedor...? Pues claro, cargando el programa, pero no con Echidna Scratch sino con código, ver en el punto 8 semáforo sonido

¿Te atreves a ...?

Realizar un programa que haga lo mismo pero con la temperatura (los valores límites de encendido y apagado de las luces dependen de la temperatura de trabajo)

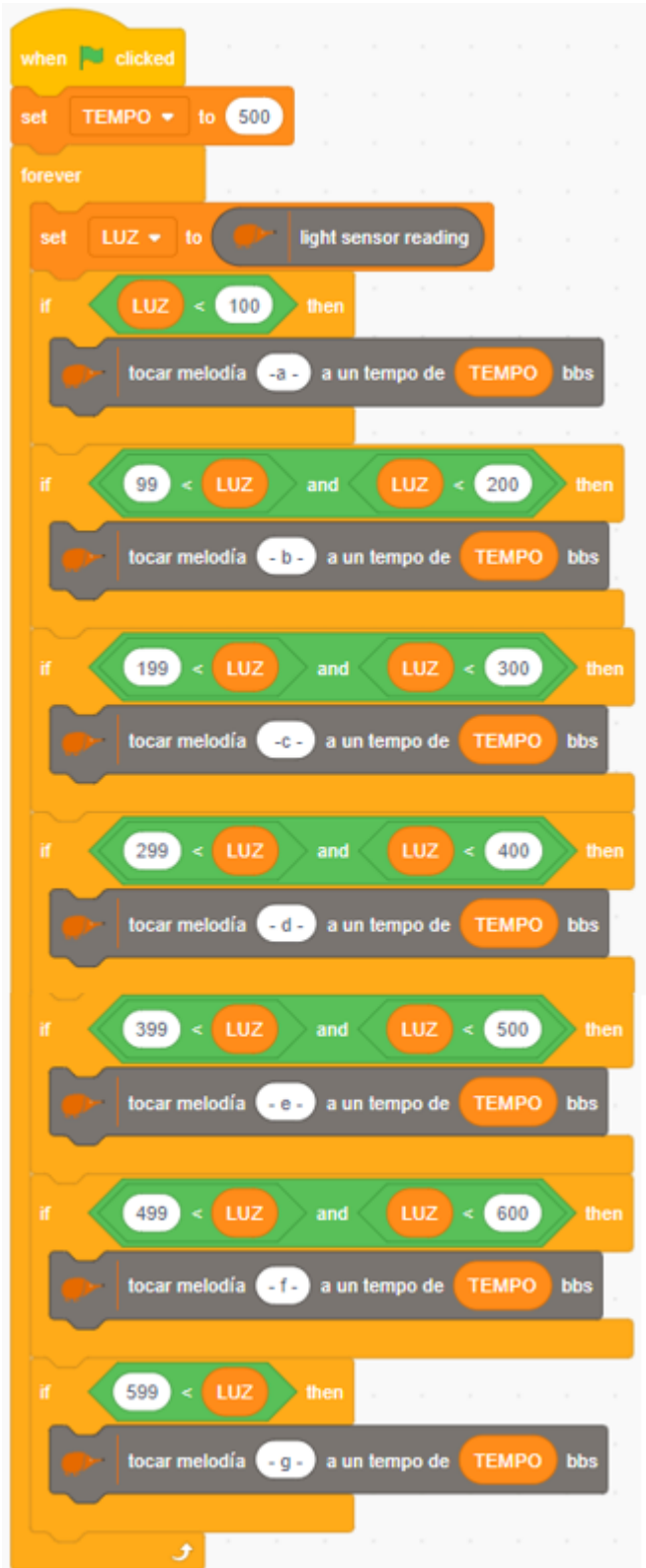
MONTAJE 6 Piano luminoso

Enunciado

Realizar un programa que suene una nota diferente según la luz

Solución en EchidnaScratch

En echidna no podemos dar las notas numéricas (en mBlock sí) tenemos que utilizar la notación midi a b c d e f g luego simplemente según un nivel de luz que toque una de esas notas



<https://www.youtube.com/embed/O59dMDvtELQ>

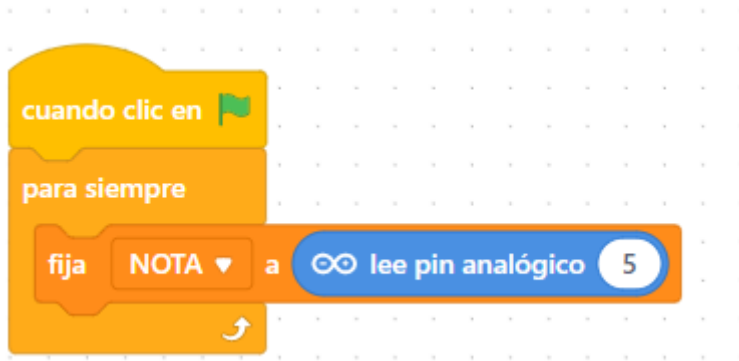


Todos los programas de este curso se encuentran en este repositorio:

<https://github.com/JavierQuintana/Echidna>

Solución con mBlock5

En el Arduino el programa es



En el objeto, el panda es



Nota: se ha incorporado una música con un tono, el C Elec Piano

El programa lo tienes aquí

<https://planet.mblock.cc/project/projectshare/3228812>

Con mBlock versión 3 ATENCIÓN ESTA VERSIÓN ES OBSOLETA NO RECOMENDADO sólo para PCs muy viejos

En esta versión no es tan sencillo ¿por qué? Porque hay cambios de escala: el LDR nuestro trabaja con valores distintos al de las notas, por lo tanto hay que hacer UN CAMBIO DE ESCALA, y esto

necesita un apartado diferente, te recomendamos ver [3.2.3.1 Cambios de escala](#)

Solución

El programa es pues el siguiente



¿por qué lo hacemos con la opción de subir a Arduino? Porque la simulación va lenta (se oye tut-tut-tut) si lo subes al Arduino reproduce el tono correctamente. [VER COMO SUBIR AL ARDUINO](#).

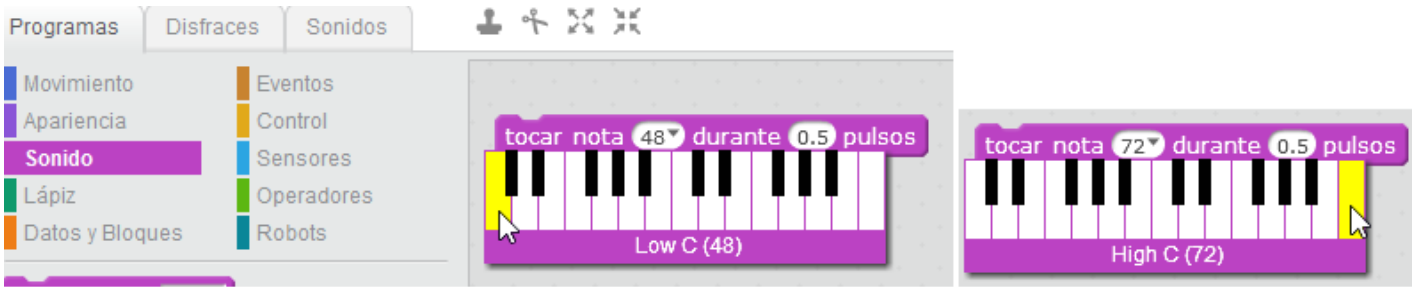
El resultado lo puedes ver en [este vídeo](#):

<https://www.youtube.com/embed/b7SSXn4q8WM>

3.2.3.1 Cambios de escala

Nos encontramos que:

- Los valores de entrada, es decir, el LDR trabaja con unos límites, que según hemos visto en [el apartado comprobar los límites](#), el nuestro va de 108 a 982, llamaremos a esta **variable de entrada X**
- Los valores de salida son los tonos, que mBlock trabaja con la norma americana (la europea es el típico Do-Re-Mi-Fa-Sol) y los americanos, son valores numéricos que van desde 48 hasta 72 (lo puedes comprobar con la instrucción “tocar nota” que está en Programas-Sonido). Llamaremos a la **variable salida de notas Y**:



Problema: ¿cómo convertimos X en Y?

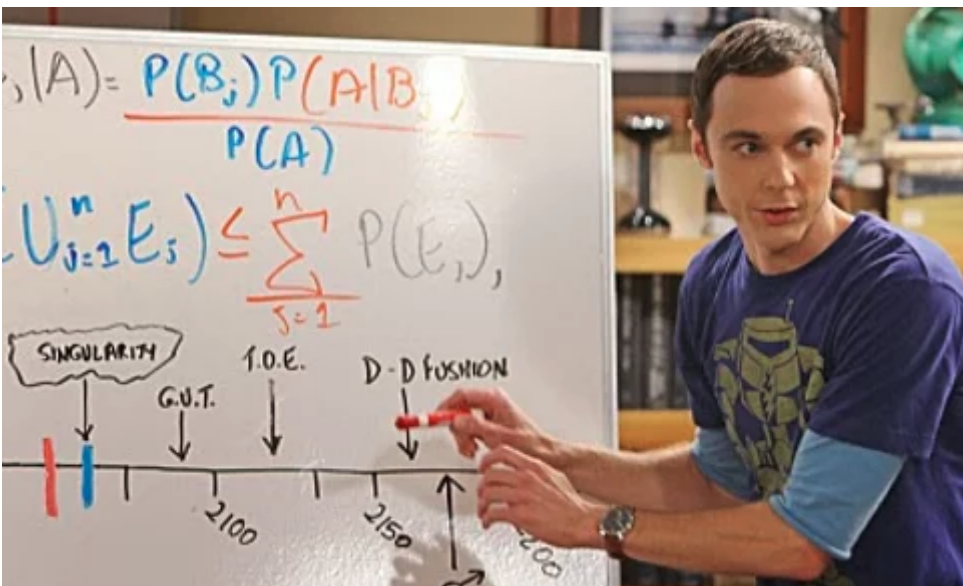
Matemáticamente es una recta con una pendiente m y una ordenada n :

$$Y = m \cdot X + n$$

Para calcular m y n tenemos que utilizar un sistema de ecuaciones dadas las condiciones límites de X e Y:

- Cuando $X = 108$ quiero que Y valga 48: $48 = m \cdot 108 + n$
- Cuando $X = 982$ quiero que Y valga 72: $72 = m \cdot 982 + n$

Ala! dos ecuaciones y dos incógnitas: calcula m y n



Buen ejercicio para los alumnos para que vean matemáticas aplicadas

No obstante, como esto lo repetiremos muchas veces los cambios de escala, hemos confeccionado [ESTA HOJA DE CÁLCULO](#) que te lo puedes descargar y facilita las cosas: Ponemos en las celdas amarillas los valores límites :



- $X1= 108$ $X2=982$
- $Y1=48$ $Y2=72$

Y nos da los valores m y n automáticamente:

A	B	C
HOJA DE CALCULO DE CAMBIO DE ESCALAS		
$y = m X + n$	CAMBIA LO AMARILLO	
Valores que tengo de x		
X1=	108	
X2=	982	
Valores que quiero de Y		
Y1=	48	
Y2=	72	
RESULTADOS		
m =	0,02745995	
n =	45,0343249	

Luego la fórmula para el cálculo de la Y (las notas) en función de la luminosidad X es:

$$Y = 0.027 * X + 45$$

MONTAJE 7 Telesketch

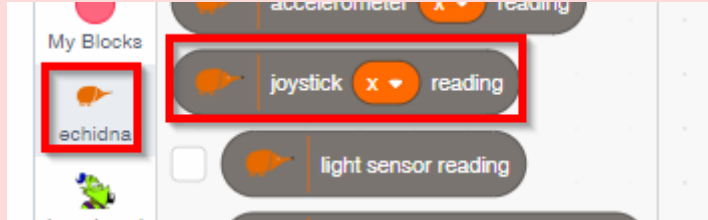
¿Qué es el Joystick?

Son dos potenciómetros, uno en el eje X y otro en el eje Y además de un pulsador digital cuando se pulsa hacia dentro.

El potenciómetro X está conectado en los pines A0 del Arduino y el del eje Y al A1 por lo tanto si has leído los capítulos de ARDUINO **sus valores varía desde 100 hasta 1024**. El pulsador está conectado al D10.

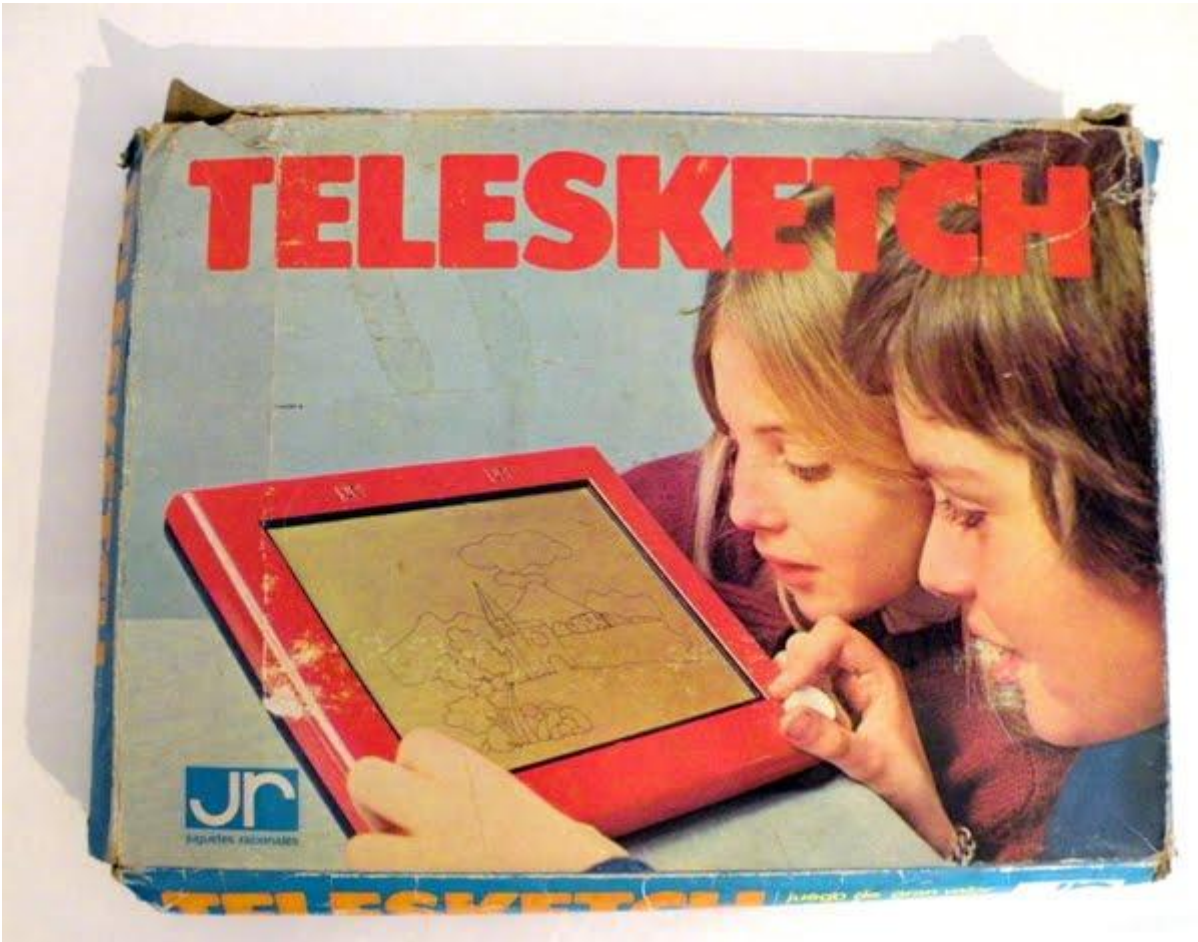


ATENCIÓN esta instrucción NO funciona, hay que utilizar las que se ponen en los ejemplos



Reto Telesketch

Si no sabes lo que es esto, no has tenido infancia [] [] [] []



Vamos a realizar el mismo ejemplo que [este vídeo de Jorge Lobo](#) pero en vez de realizado en Snap4Arduino lo vamos a hacer en EchidnaScratch y en mBlock.

https://www.youtube.com/embed/Hx5DjQw7e_U

Solución con EchidnaScratch

Simplemente vamos a considerar estas condiciones:

- Si el joystick en el **eje X** es menor que **300** es que estás inclinando el Joystick hacia la izquierda, luego el Sprite tiene que modificar su variable **decrementando** su valor, un valor fijo que lo definiremos como PASO
-

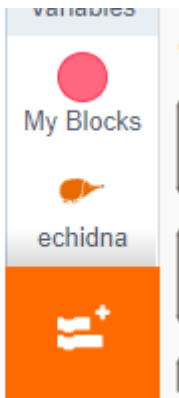


Si el joystick en el **eje X** es mayor que **700** es seguro que estás inclinando el Joystick hacia la derecha luego hay que **incrementar** el valor de la ordenada X el valor predefinido PASO

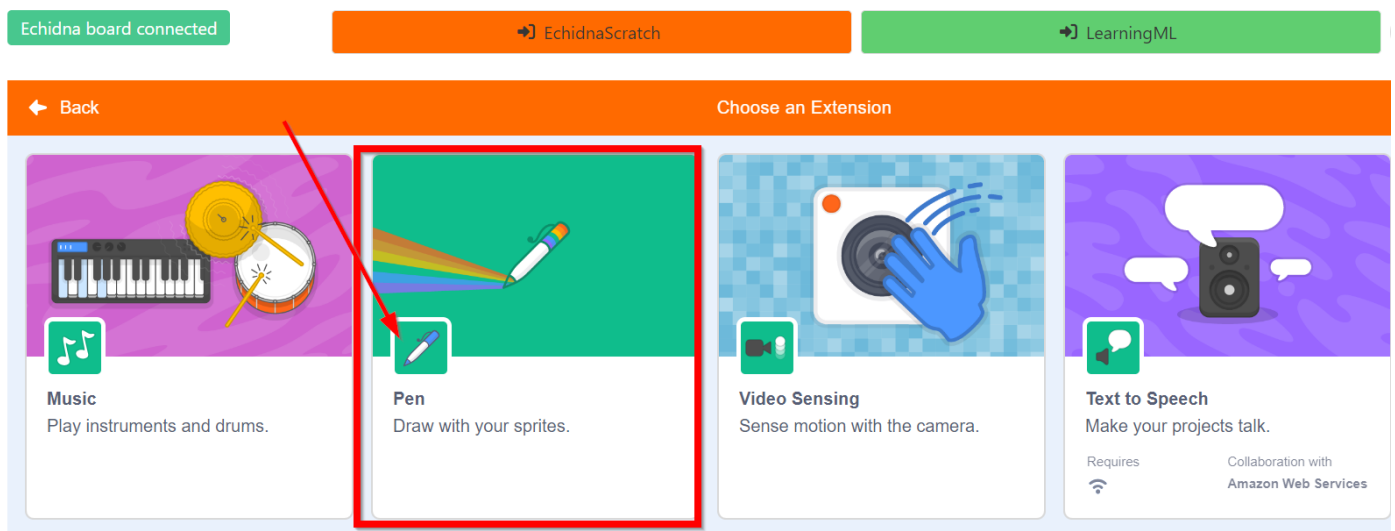
- *Idem para el **eje Y***
- **PASO** es una variable que de momento lo vamos a poner como 3, si es menor, va más preciso pero más lento, si es mayor es más rápido pero el dibujo es menos preciso.
- Los botones SR y SL nos servirán para subir y bajar el lápiz y borrar

¿Lápiz ? ¿Qué lápiz?

Lápiz es una extensión para poder dibujar. Para instalar una extensión pulsa aquí

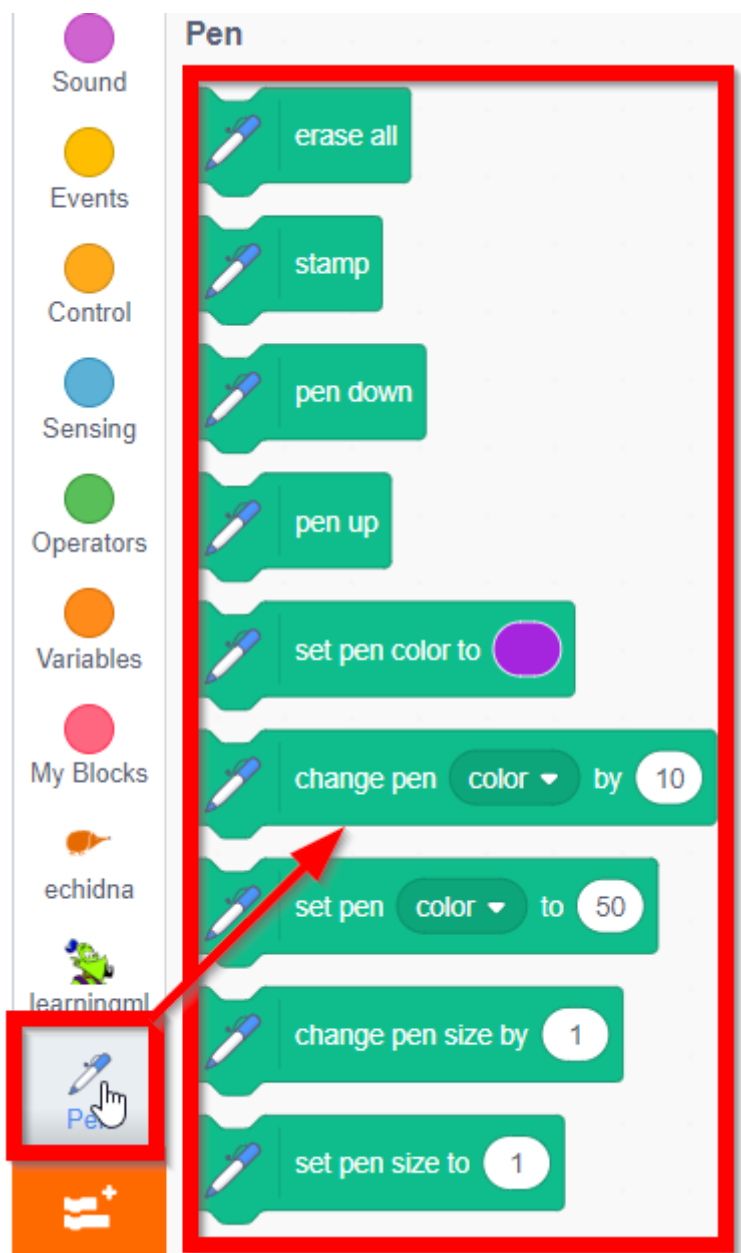


E instalamos esta extensión



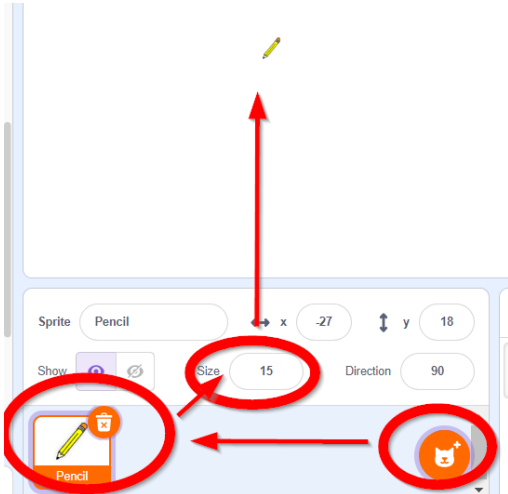


Y se instalan nuevas instrucciones para dibujar



Código con EchidnaScratch

Recomendamos cambiar el sprite del gato por otro más apropiado, por ejemplo el lápiz y bajarlo a un tamaño 15



El programa :

- Inicia las variables que hemos explicado
- En el bucle, las condicionales hacen que se incremente o decremente las variables X e Y la cantidad definida PASO
- El botón SR
 - si se pulsa, pinta
 - si se suelta no pinta
- El botón SL borra todo
- Mover el sprite según las coordenadas X e Y





Todos los programas de este curso se encuentran en este repositorio:

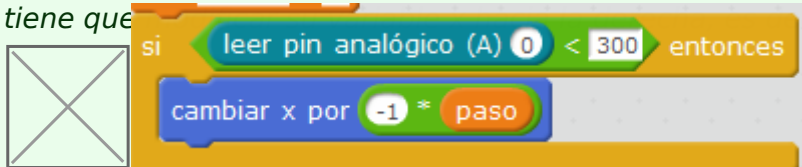
<https://github.com/JavierQuintana/Echidna>

https://www.youtube.com/embed/a27fEoP_A2w

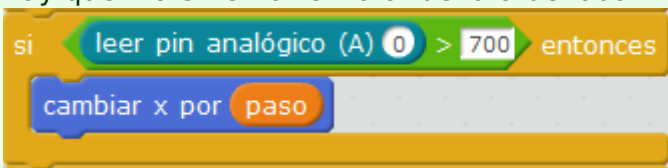
Solución con mBlock

En mBlock no hay instrucciones especiales para Echidna, hay que hablar pues de A0 para el eje X y A1 para el eje Y por lo tanto las nuevas condiciones son:

Si A0 es menor que 300 es que estás inclinando el Joystick hacia la izquierda, luego el Sprite tiene que ir decrementando su valor



Si A0 es mayor que 700 es seguro que estás inclinando el Joystick hacia la derecha luego hay que incrementar el valor de la ordenada X



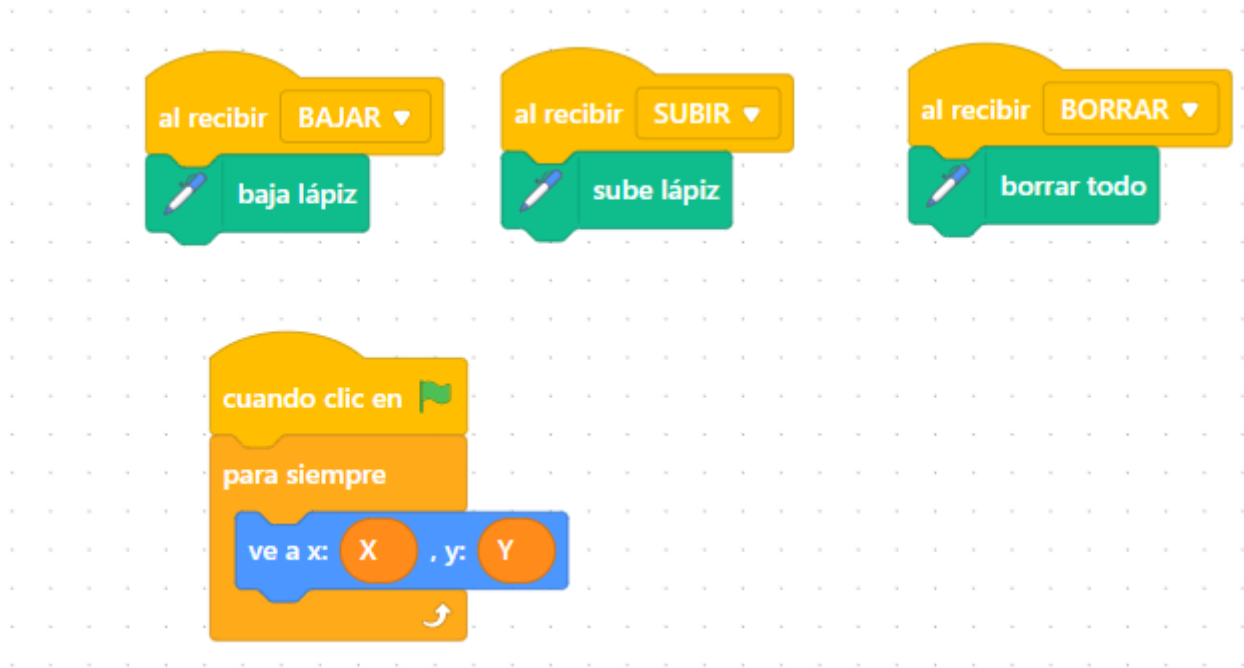
Para los valores del eje Y es igual, cambiando A0 por A1

Programa en mBlock

Primero pondremo un objeto lápiz y añadiremos la extensión LAPIZ esto se entiende mejor con un vídeo

<https://www.youtube.com/embed/xft-6Nz3yzE>

Pero no hagas la programación de hacer el cuadrado. Sino este



Donde BAJAR, SUBIR Y BORRAR SON MENSAJES y X e Y son variables globales a todos los objetos.
En el arduino el programa será



El programa te lo puedes descargar aquí <https://planet.mblock.cc/project/3228849>

VA MUY LENTO El resultado lo puedes ver en [este vídeo](#):

<https://www.youtube.com/embed/jzyd5cPb2-Y>

Por curiosidad puedes ver el mismo programa en **Snap4Arduino** en este vídeo:

<https://www.youtube.com/embed/j1lsYq6X5-U>

Puedes ver que es mucho **más rápido** que mBlock.

Mapeo

¿Qué es eso de "mapeo"?

En la jerga robótica, dicho pronto y mal pero para que se entienda, mapear significa hacer un **cambio de escala**

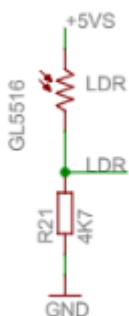
¿Cuándo se dan esas situaciones?

SITUACION A : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida digital PWM de Arduino que va de 0-255

SITUACION B : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 e interpretarlo en sus valores de voltios. Si suponemos que la placa se alimenta a 5V la variable de salida irá desde 0 a 5V

SITUACION C : Queremos leer el valor de un LDR, que tapándolo nos da 917 e iluminándolo al máximo es 1023, lo queremos copiar en una salida digital PWM, o sea que la salida va desde 0 a 255

Nota: El mínimo de 917 (puede ser otro número, es un valor experimental) es debido a que los LDR van montados en un divisor de tensión como el de la figura, y la resistencia de abajo, siempre se queda algo de tensión



SITUACION D : Queremos según el valor de un joystick conectado a las entradas analógicas de un Arduino (esto pasa en Echidna) se representen en la pantalla de Scratch



2*220 por 2*180, es decir

- Eje X : el potenciómetro (vamos a llamarlo *potx*) va de 0 a 1023 y la salida (*ejex*) va de - 220 a 220
- Eje Y : el potenciómetro (vamos a llamarlo *poty*) va de 0 a 1023 y la salida (*ejey*) va de - 180 a 180

SITUACION E: Ídem pero no con el potenciómetro, sino con el acelerómetro (vamos a llamarlo *acel*) que va 250 a 500

SITUACION F : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida de un servo, por lo tanto lo que necesita es un ángulo que va de 0-180

SITUACION G : Ídem que F pero una raspberry por lo tanto GPI va de 0-65.535

¿Cómo se consigue mapear?

- Si programas con código ArduinoIDE, tienes la instrucción **map**
- Si no tienes map, por ejemplo, programas con bloques gráficos tipo Scratch, lo tienes que hacer a mano
 - ¿Cómo? Con la ecuación de una recta

Para entendernos :

- **X** será el valor de entrada que tiene unos valores límites **X₁** e **X₂**
- **Y** es la variable de salida que queremos y que tiene otros valores límites **Y₁** e **Y₂**

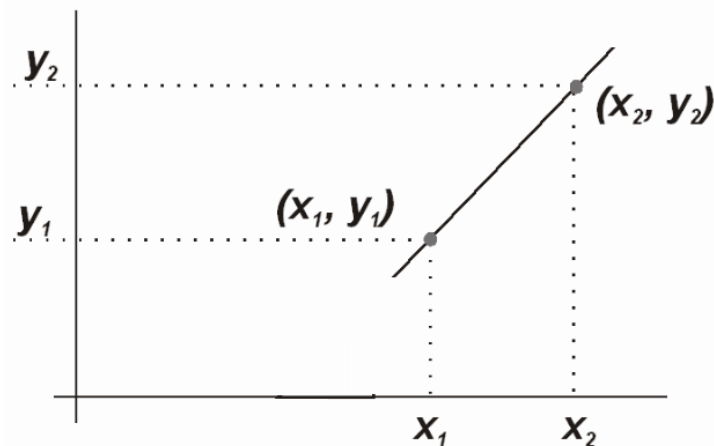
Luego y tiene esta ecuación :

$$y = y_1 + m * (x - x_1)$$

donde m es

$$m = \frac{x_2 - x_1}{y_2 - y_1}$$

Gráficamente



¿Me lo puedes hacer para cada situación anterior?

Si claro:

SITUACION A : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida digital PWM de Arduino que va de 0-255

- Límites de las variables :
 - X de 0-1023
 - Y de 0-255
- Con la instrucción map : $Y = \text{map}(X, 0, 1023, 0, 255);$
- Sin la instrucción map $Y = 0.25 * X$ pues $255/1023 = 0.25$ también podemos escribir $Y = X/4$

SITUACION B : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 e interpretarlo en sus valores de voltios. Si suponemos que la placa se alimenta a 5V la variable de salida irá desde 0 a 5V

- Límites de las variables :
 - X de 0-1023
 - Y de 0-5
- Con la instrucción map : $Y = \text{map}(X, 0, 1023, 0, 5);$
- Sin la instrucción map $Y = 0.0048 * X$ pues $5/1023 = 0.0048$ o también podemos escribir $Y = X/204$ que queda mejor pues $1023/5=204$ aprox.

SITUACION C : Queremos leer el valor de un LDR, que tapándolo nos da 917 e iluminándolo al máximo es 1023, lo queremos copiar en una salida digital PWM, o sea que la salida va desde 0 a 255

- Límites de las variables :
 - X de 917-1023
 - Y de 0-255
- Con la instrucción map : $Y = \text{map}(X, 917, 1023, 0, 255);$
- Sin la instrucción map $Y = 2.4 * X$ pues $255/(1023-917) = 2.4$

SITUACION D : Queremos según el valor de un joystick conectado a las entradas analógicas de un Arduino (esto pasa en Echidna) se representen en la pantalla de Scratch 2*220 por 2*180, es decir

- Eje X : el potenciómetro (vamos a llamarlo *potx*) va de 0 a 1023 y la salida (*ejex*) va de -220 a 220
- Eje Y : el potenciómetro (vamos a llamarlo *poty*) va de 0 a 1023 y la salida (*ejey*) va de -180 a 180

- EJEX
 - Límites de las variables :
 - *potx* de 0-1023
 - *ejex* de -220 a +220
 - Con la instrucción map : $\text{ejex} = \text{map}(\text{potx}, 0, 1023, -220, 220);$
 - Sin la instrucción map $\text{ejex} = -220 + 0.43 * \text{potx}$ pues $(220 - (-220))/1023 = 0.43$
- EJY
 - Límites de las variables :
 - *poty* de 0-1023
 - *ejey* de -180 a +180
 - Con la instrucción map : $\text{ejey} = \text{map}(\text{poty}, 0, 1023, -180, 180);$
 - Sin la instrucción map $\text{ejey} = -180 + 0.35 * \text{poty}$ pues $(180 - (-180))/1023 = 0.35$

SITUACION E: Ídem pero no con el potenciómetro, sino con el acelerómetro (vamos a llamarlo *acel*) que va 250 a 500

- EJEX
 - Límites de las variables :
 - acelerómetro *acel* de 250-500
 - *ejex* de -220 a +220
 - Con la instrucción map : $\text{ejex} = \text{map}(\text{acel}, 250, 500, -220, 220);$



- Sin la instrucción map ejey = $-220 + 1.76 \cdot (\text{acel} - 250)$ pues $(220 - (-220)) / (500 - 250) = 1.76$
- EJEY
 - Límites de las variables :
 - acelerómetro acel de 250-500
 - ejeY de -180 a +180
 - Con la instrucción map : ejey = map(acel, 250 500, -180, 180);
 - Sin la instrucción map ejey = $-180 + 1.44 \cdot (\text{acel} - 250)$ pues $(180 - (-180)) / (500 - 250) = 1.44$

SITUACION F : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida de un servo, por lo tanto lo que necesita es un ángulo que va de 0-180

- Límites de las variables :
 - X de 0-1023
 - Y de 0-180
- Con la instrucción map : $Y = \text{map}(X, 0, 1023, 0, 180);$
- Sin la instrucción map $Y = 0.17 \cdot X$ pues $180/1023 = 0.17$ también podemos escribir $Y = X/5.7$ pues $1023/180 = 5.7$

SITUACION G : Idem que F pero una raspberry por lo tanto GPI va de 0-65535

- Límites de las variables :
 - X de 0-65535
 - Y de 0-180
- Con la instrucción map : $Y = \text{map}(X, 0, 65535, 0, 180);$
- Sin la instrucción map $Y = 0.00274 \cdot X$ pues $180/65535 = 0.00274$ pero es más cómodo al revés $Y = X/364$ pues $65535/180 = 364$

MONTAJE 8 Comebichos

No podemos dejar el Joystick sin hacer un videojuego !!



El siguiente **RETO** es: *Mover un Sprite "Bat" con el Joystick para atrapar un Beetle*

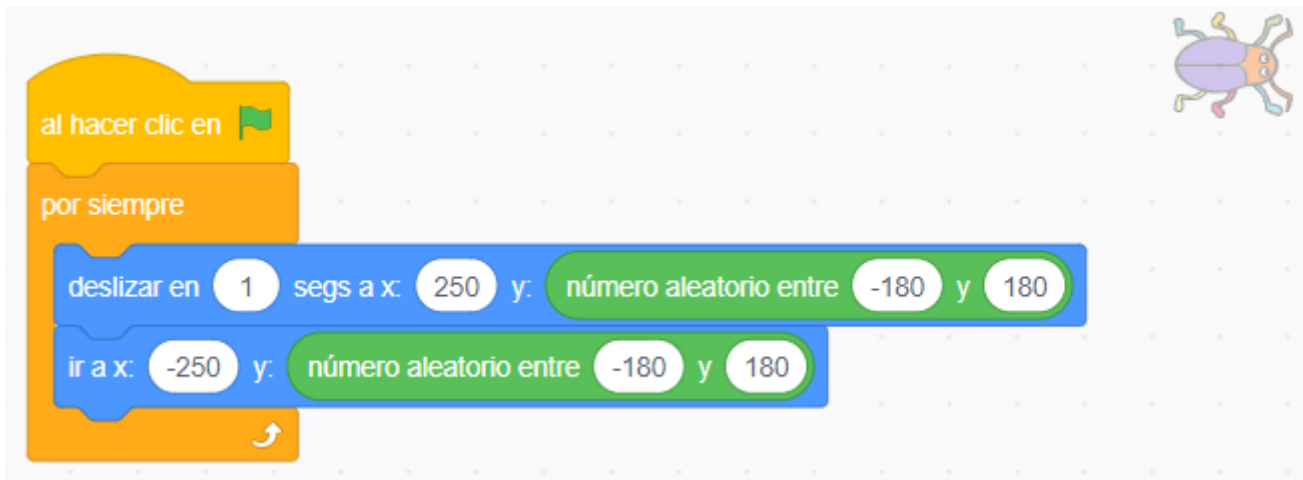
- Bat se tiene que mover con el joystick por toda la pantalla.
- Beetle se mueve horizontalmente desde el borde derecho hasta el borde izquierdo, y cuando llega al izquierdo, vuelve a aparecer en el derecho (y aleatoriamente desde cualquier altura)



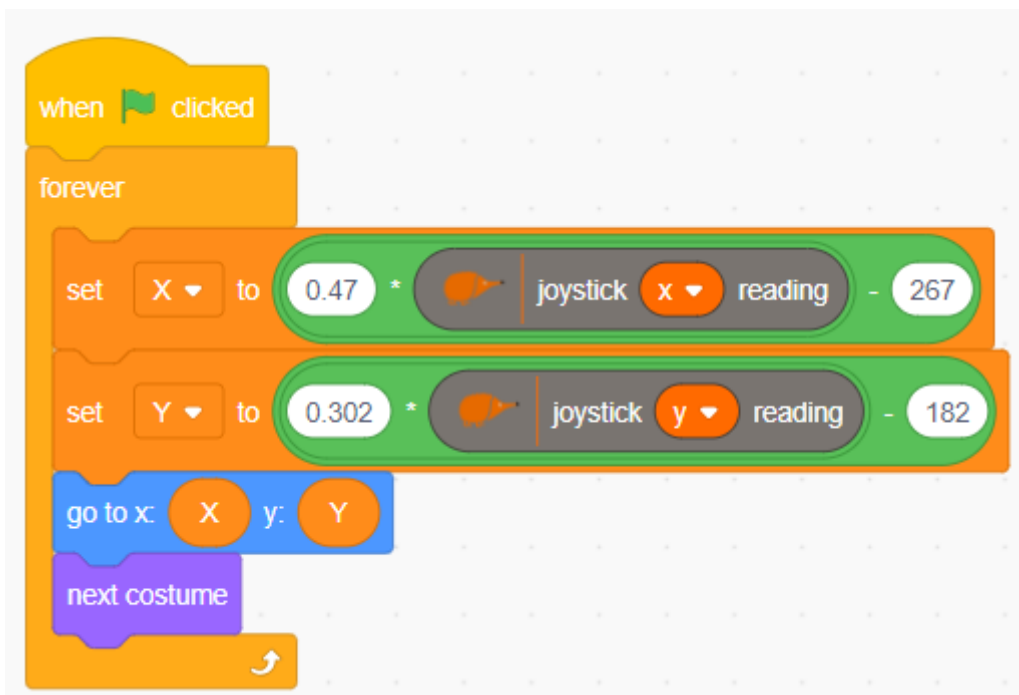
Solución con EchidnaScratch

Añadimos los Sprite Beetle y Bat y borramos el gato

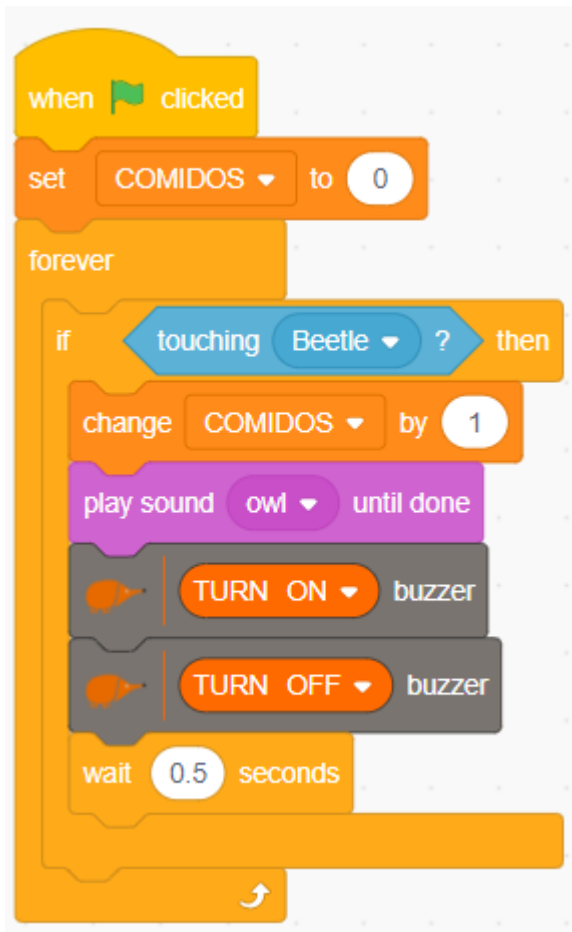
Al sprite BEETLE vamos a moverlo aleatoriamente con este script



El sprite BAT se moverá según la posición del JOYSTICK pero **mapeadas** es la SITUACION D de la página que has visto [mapeo](#) algo retocadas.



y le añadimos el siguiente código al BAT para que cuente los bichos comidos. Previamente crear una variable COMIDOS



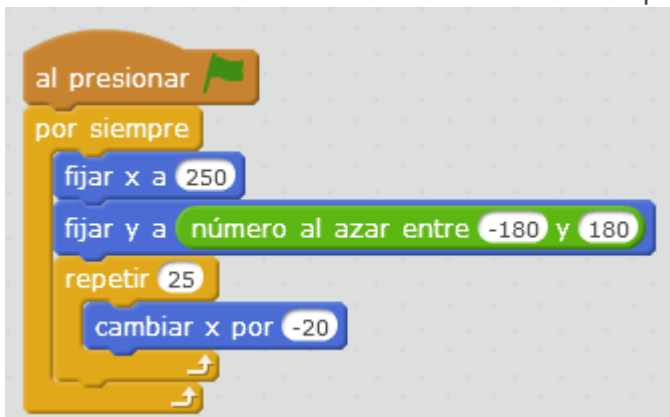
Todos los programas de este curso se encuentran en este repositorio:

<https://github.com/JavierQuintana/Echidna>

<https://www.youtube.com/embed/kDYtr4lwszE>

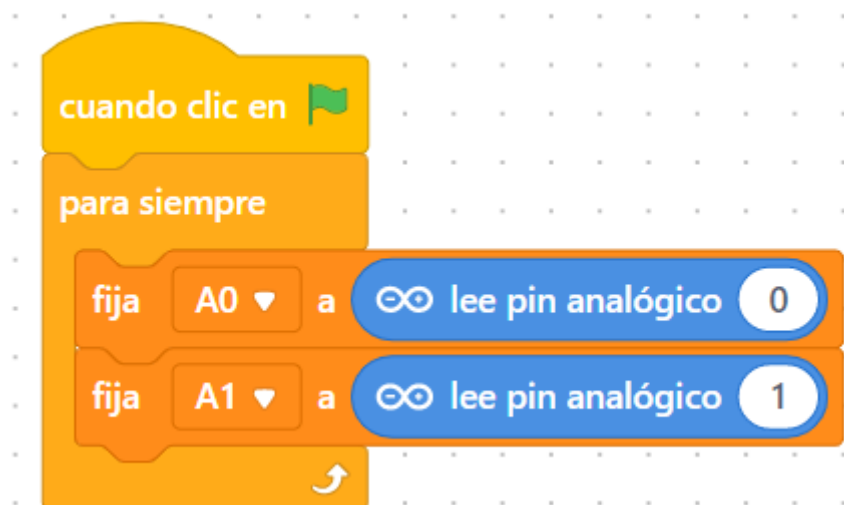
Solución con mBlock

El bicho Bettle lo haremos mover bastante rápido y al azar en el eje Y para que lo tenga difícil Bat:



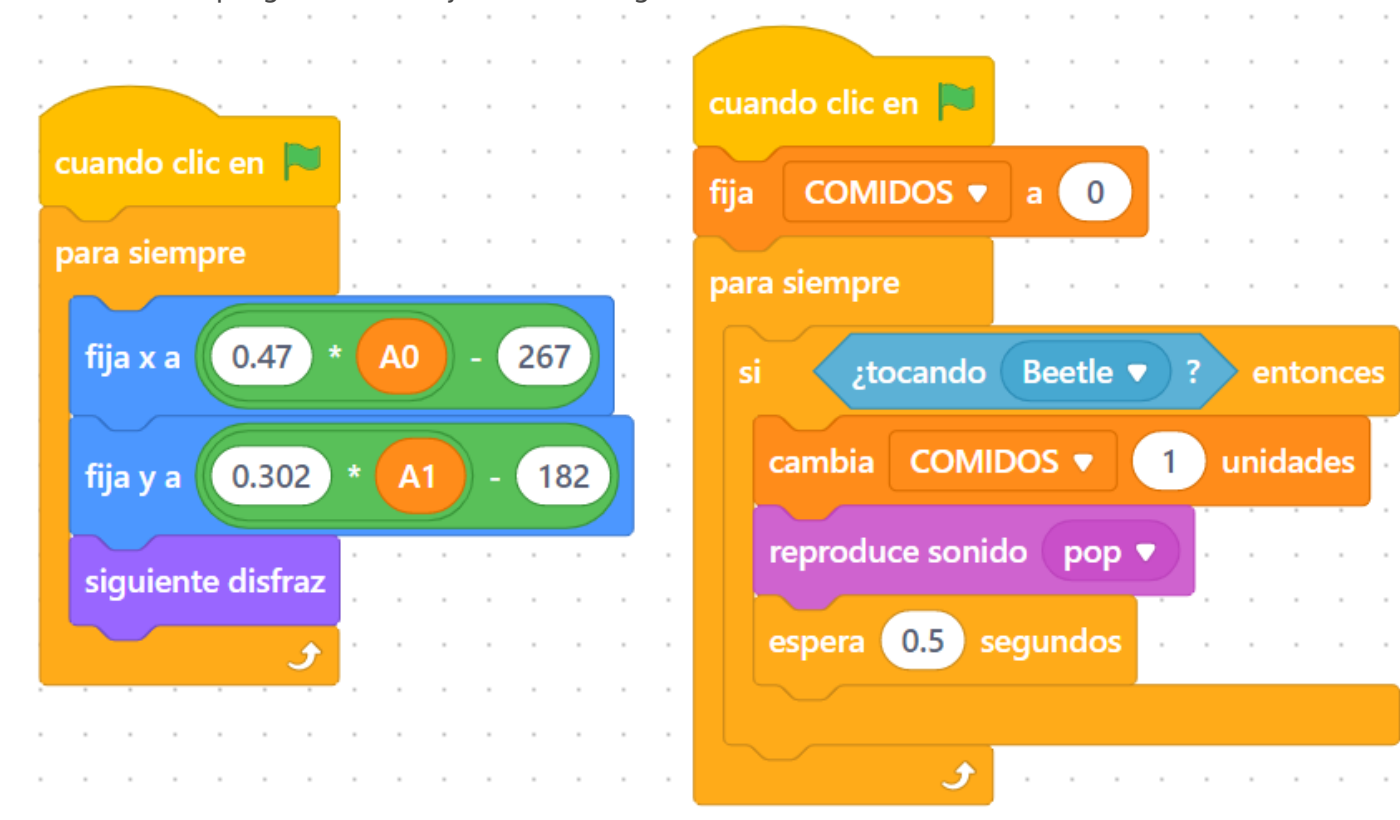


Y bat tiene que moverse con el joystick, que le pasaremos las variables A0 y A1. Por lo tanto el Arduino tiene este programa:



Además hemos añadido la puntuación y el cambio de disfraz para que parezca que aletea:

Por lo tanto el programa del objeto murciélago es



El programa completo lo puedes descargar aquí

<https://planet.mblock.cc/project/projectshare/3230393>

El resultado es:

<https://www.youtube.com/embed/VERfepEkNv8>

Lo sé soy bastante malo !!



MONTAJE 9 SPRITEVOLADOR

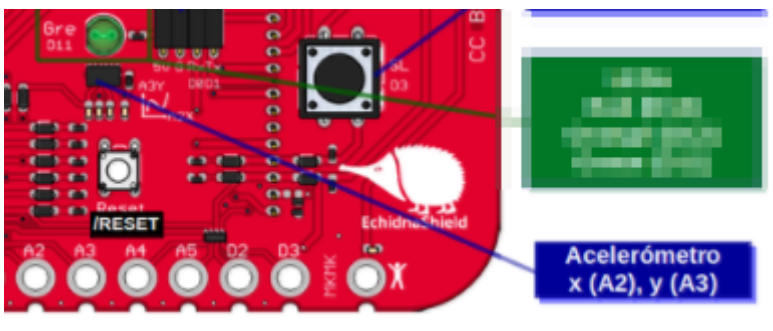
¿Qué es un acelerómetro?

El acelerómetro tiene el mismo efecto que el Joystick, nos da valores en el eje X o eje Y, en el caso del Joystick lo daba la inclinación de la palanca, en este caso lo da la inclinación del mismo Echidna.

Está conectado a los pines analógicos:

- A2 nos da la inclinación en el eje X
- A3 nos da la inclinación en el eje Y

Los valores van desde 250 hasta 500



MONTAJE 9 SPRITEVOLADOR

Vamos a realizar un sencillo videojuego: Mover un **sprite volador**, el movimiento de un sprite con el acelerómetro y esquivando edificios que se mueven de derecha a izquierda dando la sensación de que el sprite vuela.

El reto te lo complicamos con dos premisas:

- Modificar disfraces del sprite
- Reutilizar sprites externos desde [Scratch](#)

Este programa tienes tres pasos

1. **Sprite volador moviéndose según acelerómetro**

1. Le añadimos un disfraz de explosión reutilizando de otro Sprite "sun"

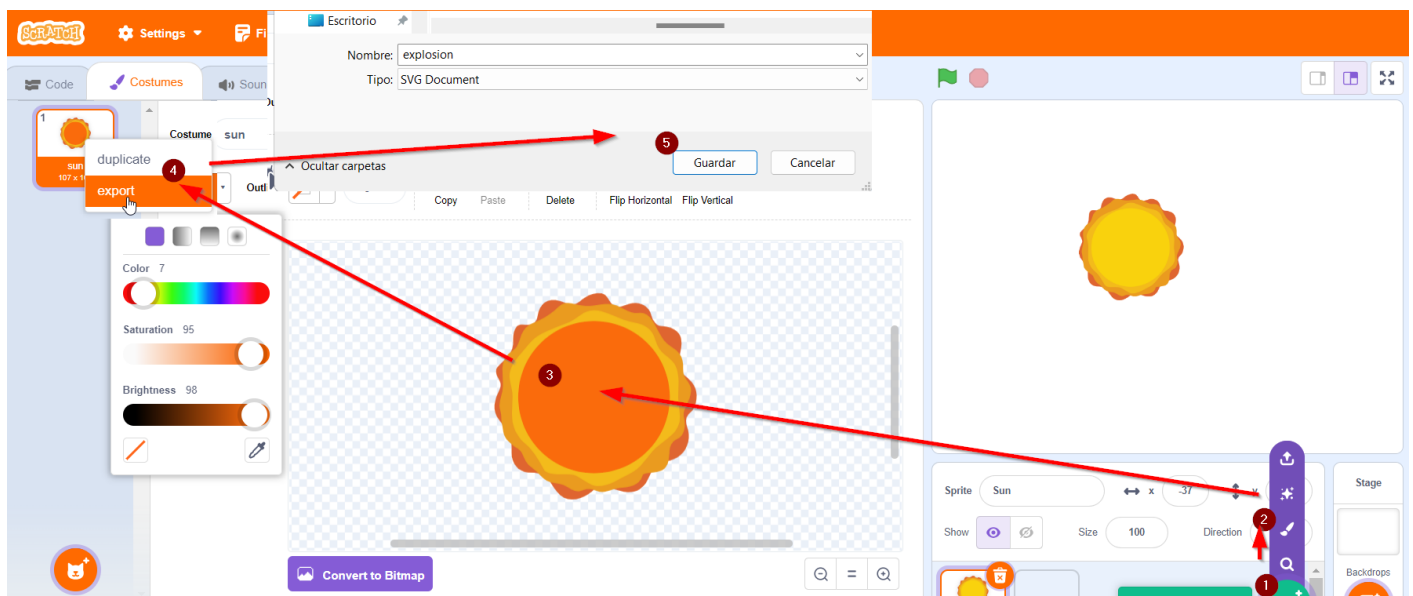
2. Movimiento según acelerómetro
2. **Sprite edificios**
 1. Exportación de sprites externos de otro autor
 2. Importación a nuestro proyecto
3. **Muerte del sprite volador si toca edificio.** Aquí es el código propio del juego

A SOLUCIÓN CON ECHIDNASCATCH

A1 Sprite PARROT moviéndose según acelerómetro

A1.1 Añadir un disfraz al PARROT de explosión.

1. Quitamos el sprite del gato y añadimos el sprite **SUN**
2. Entramos en edición del sprite
3. Lo tuneamos un poco para que parezca una explosión (simplemente pintar de rojo el centro)
4. Lo exportamos
5. Guardamos con explosion.svg

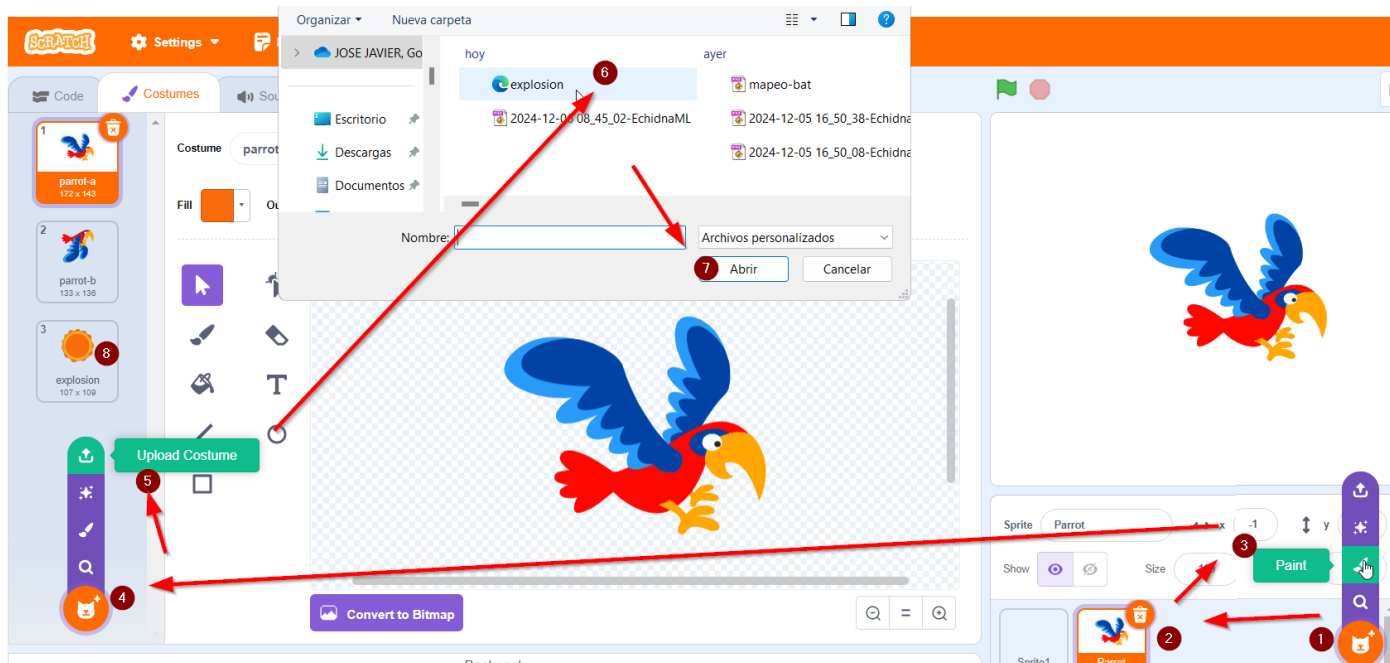


Ahora añadimos el Sprite PARROT y al sprite PARROT le ponemos el disfraz de explosión:

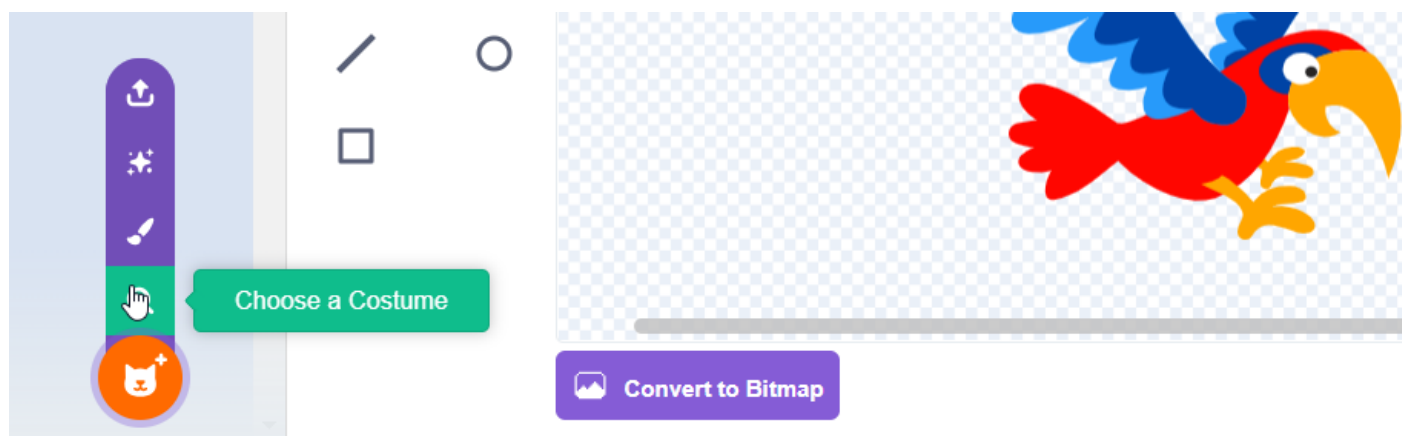
1. Añadimos el sprite PARROT
2. Borramos el anterior SUN ya no lo necesitamos
3. Entramos en Edición del sprite
4. Menú del disfraz (costume)
5. Upload costume
6. elegimos el fichero creado anteriormente "explosion.svg"
7. Abrimos



8. Ya tenemos un disfraz añadido al Parrot. Clica en el primer disfraz pues cuando vuelvas a la ventana de código verás el de explosión si no lo haces.

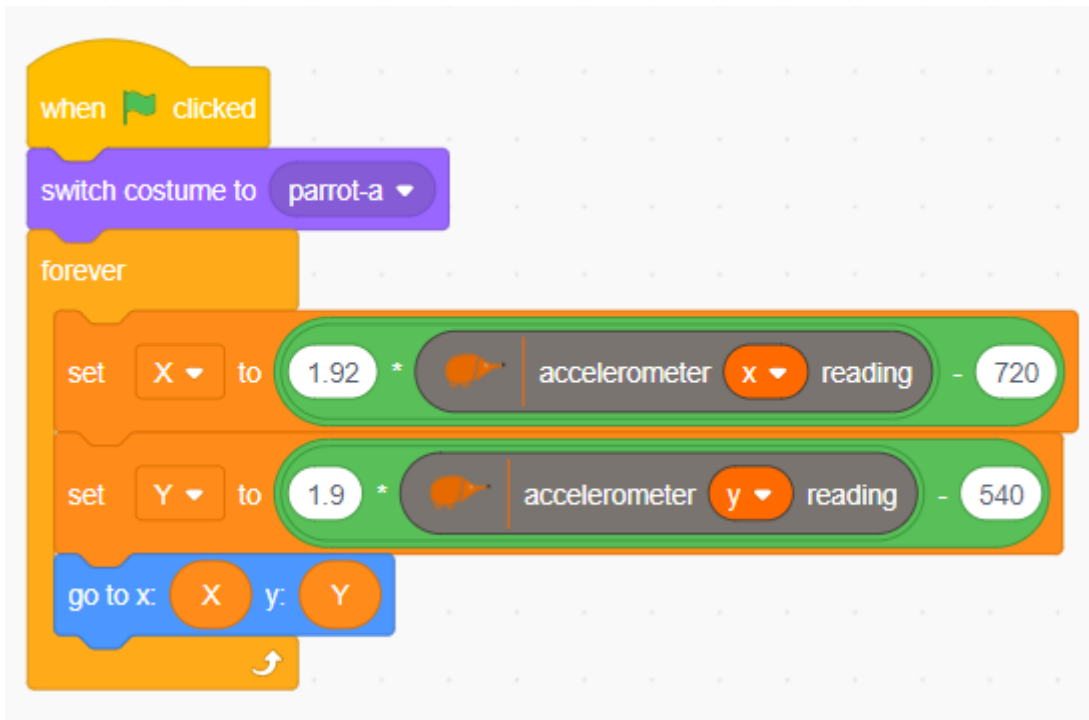


Otra manera más fácil sería simplemente añadir un disfraz de los sprite por defecto y en paz



A1.2 Movimiento del sprite PARROT

Vamos a hacer que el pájaro se mueva según el acelerómetro: (Ver al final DETALLES MAPEO)



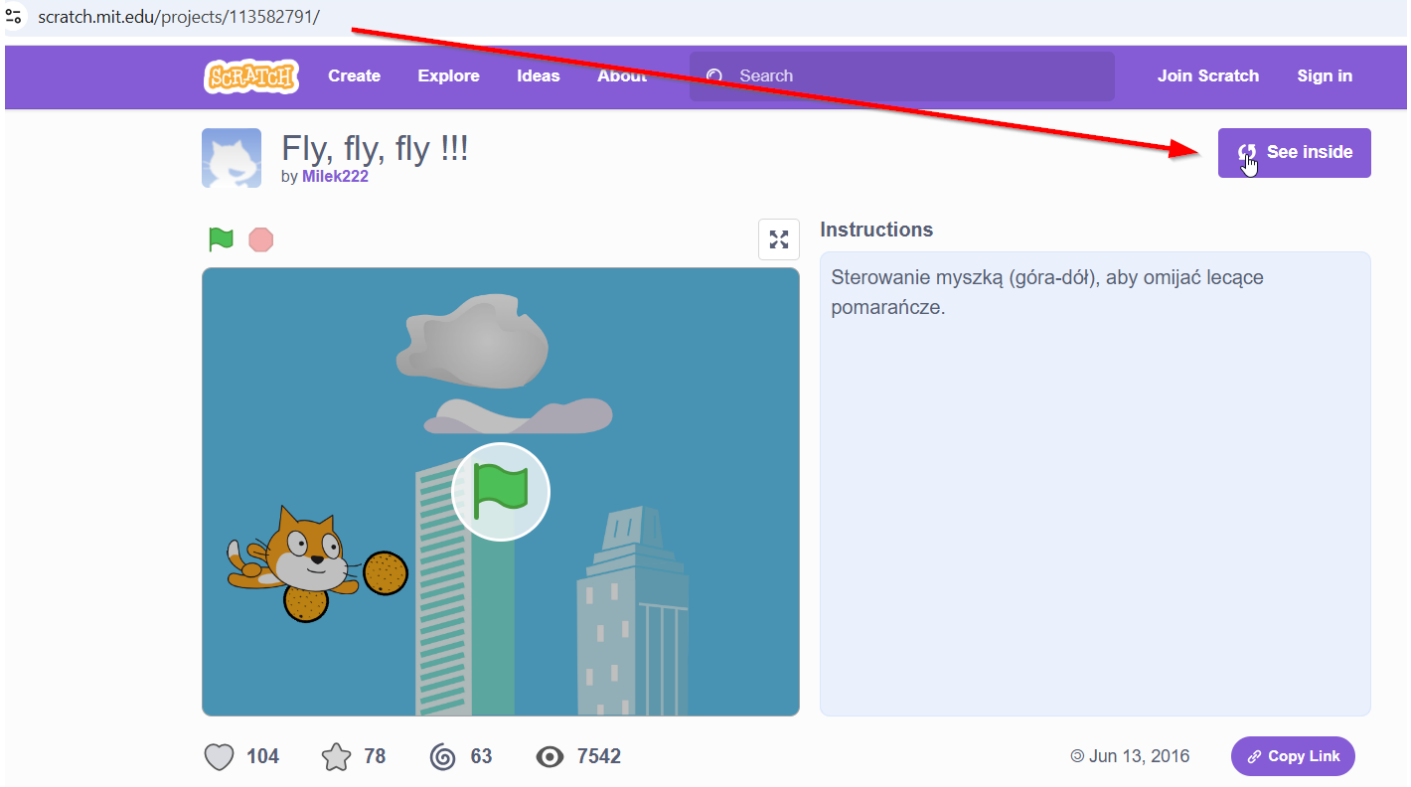
A2 Sprite EDIFICIO moviéndose de derecha a izquierda

Dará sensación al sprite pájaro que está volando.

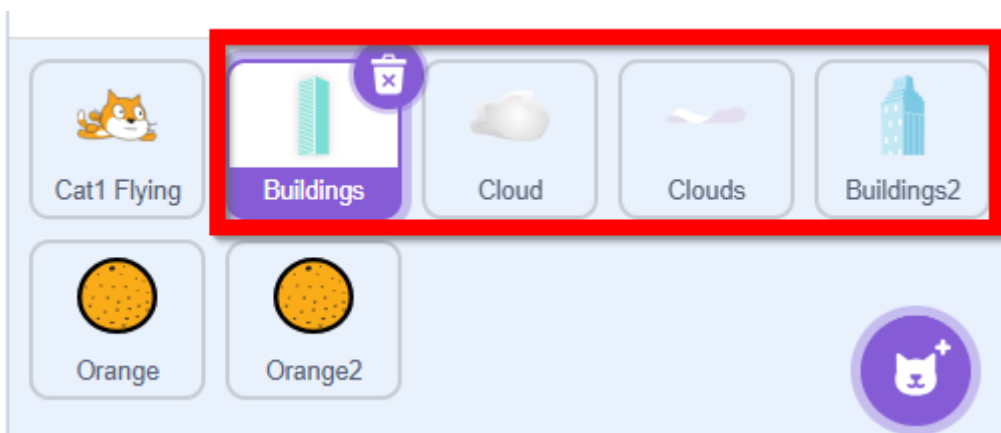
A21 Exportación de Sprites externos

Vamos a ser vagos: Nos vamos a copiar de alguien externo:

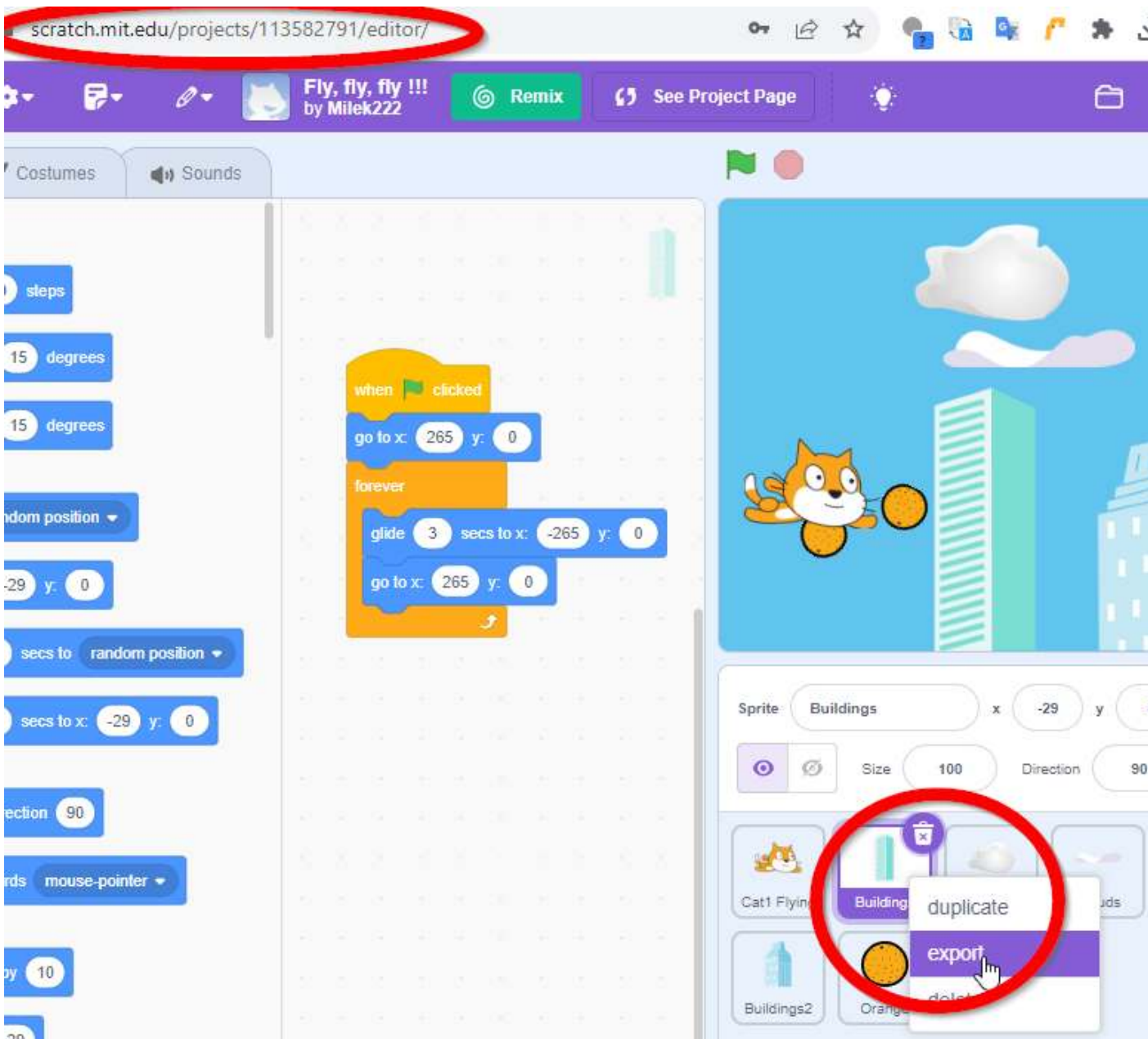
Entramos en este proyecto FLY de scratch <https://scratch.mit.edu/projects/113582791/> entramos en su código y **exportamos** los edificios, nubes, etc...



Una vez dentro del editor **EXPORTAMOS ESTOS SPRITES**



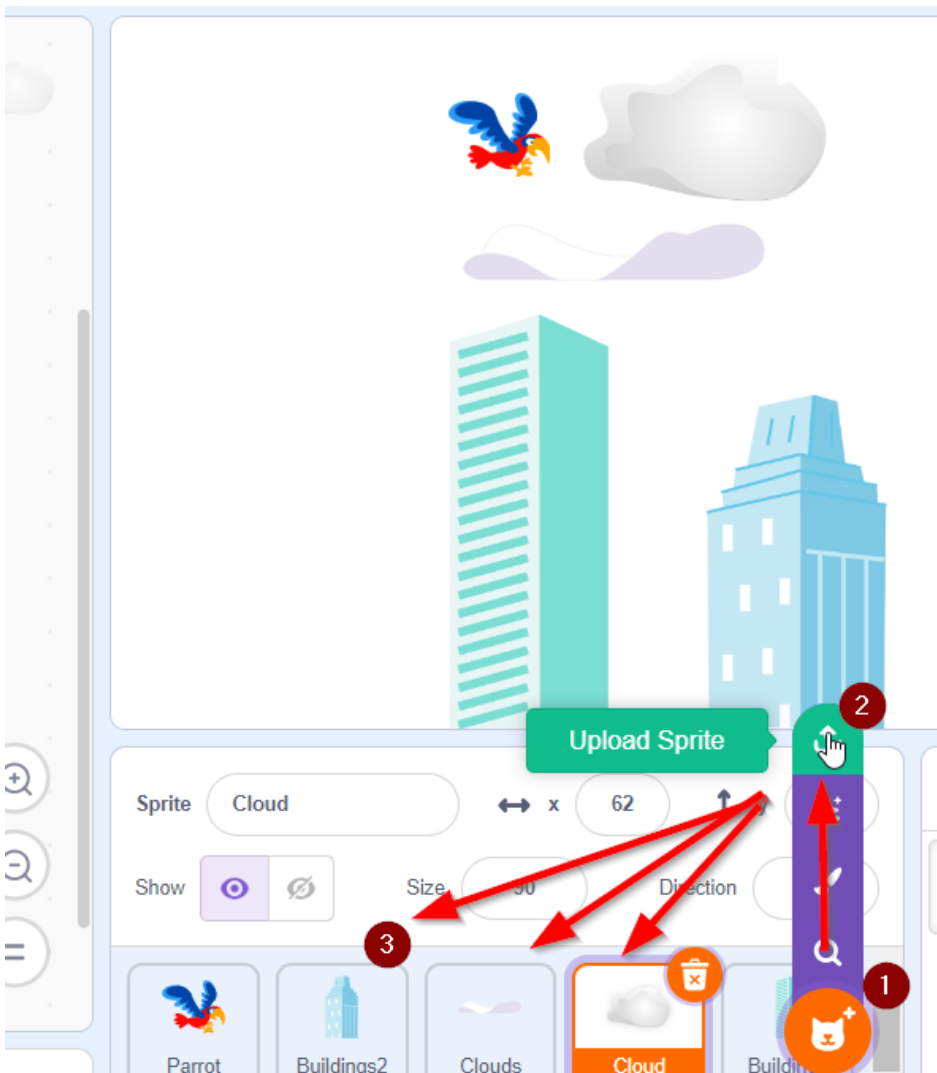
Simplemente botón derecho y **EXPORTAR**



Y guardamos esos ficheros generados **Buildings2.sprite3 Clouds.sprite3 Cloud.sprite3 Buildings.sprite3**

A22 Importación de los Sprites externos

1. Entramos en Añadir Sprite
2. Upload
3. Elegimos los ficheros exportados anteriormente en A21



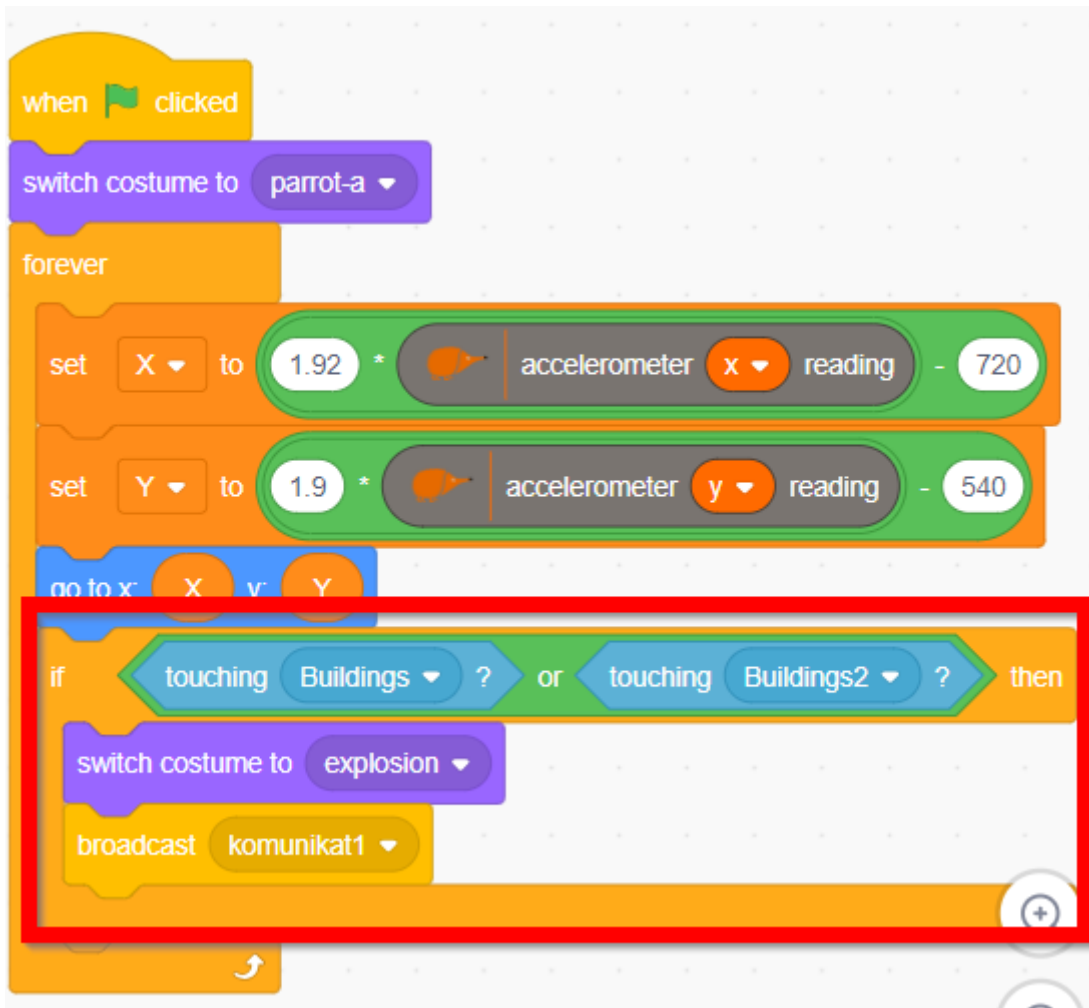
INCREIBLE

nos importa el Sprite **CON SU CÓDIGO INCLUIDO** si le das a la Bandera verás que el efecto sin apenas realizar trabajo

A3 Muerte del PARROT

Ya lo tenemos casi, añadimos al bucle de PARROT un condicional:

- Si PARROT toca BUILDINGS o BUILDINGS2 entonces
 - Cambia de disfraz **EXPLOSION**
 - Envía mensaje **komunikat1** que es el que utiliza el autor de los edificios para parar su movimiento, (debe de ser ruso su autor)



<https://www.youtube.com/embed/aMY6qiib-Bc>

MEJORAS

Ya sé que colocando el helicóptero siempre arriba, siempre ganas!!

HAZ QUE SOLO SI TOCA BUILDING1 MUERE, PERO SI TOCA A BUILDING2 QUE GANE UN PUNTO de esta manera ya no se gana poniendo el helicóptero arriba del todo

<https://giphy.com/embed/f79OYWh5uwlfK>

Todos los programas de este curso se encuentran en este repositorio:

<https://github.com/JavierQuintana/Echidna>

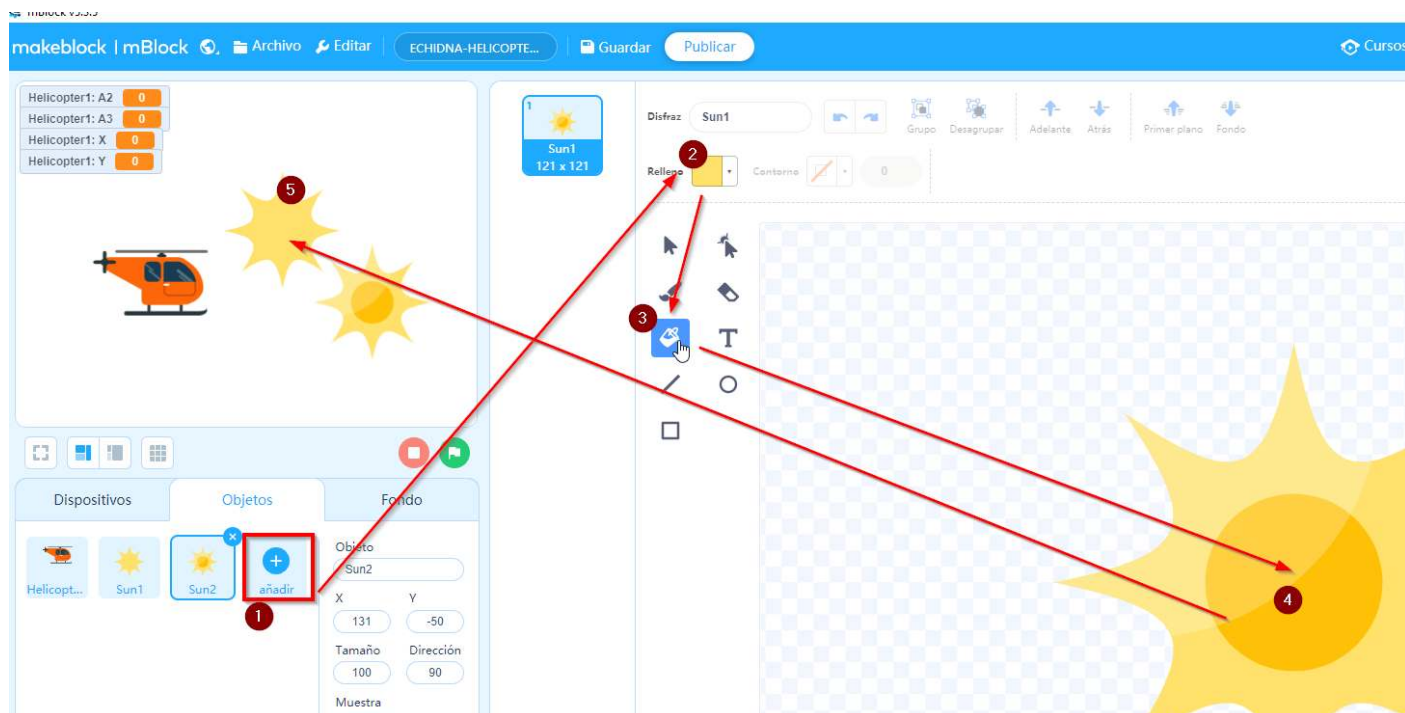
B SOLUCIÓN CON mBlock

B1 Sprite helicóptero moviéndose según acelerómetro

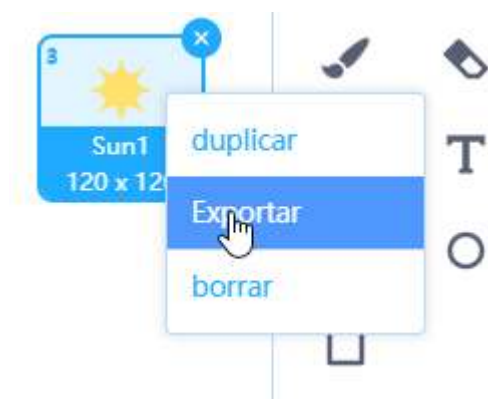
B11 Añadir un disfraz al helicóptero de explosión.

Vamos a añadir un objetos: helicóptero de la biblioteca- Le vamos a añadir un disfraz explosión

1.- Añade un objeto sun, y borra el dibujo interior, y así parece una explosión



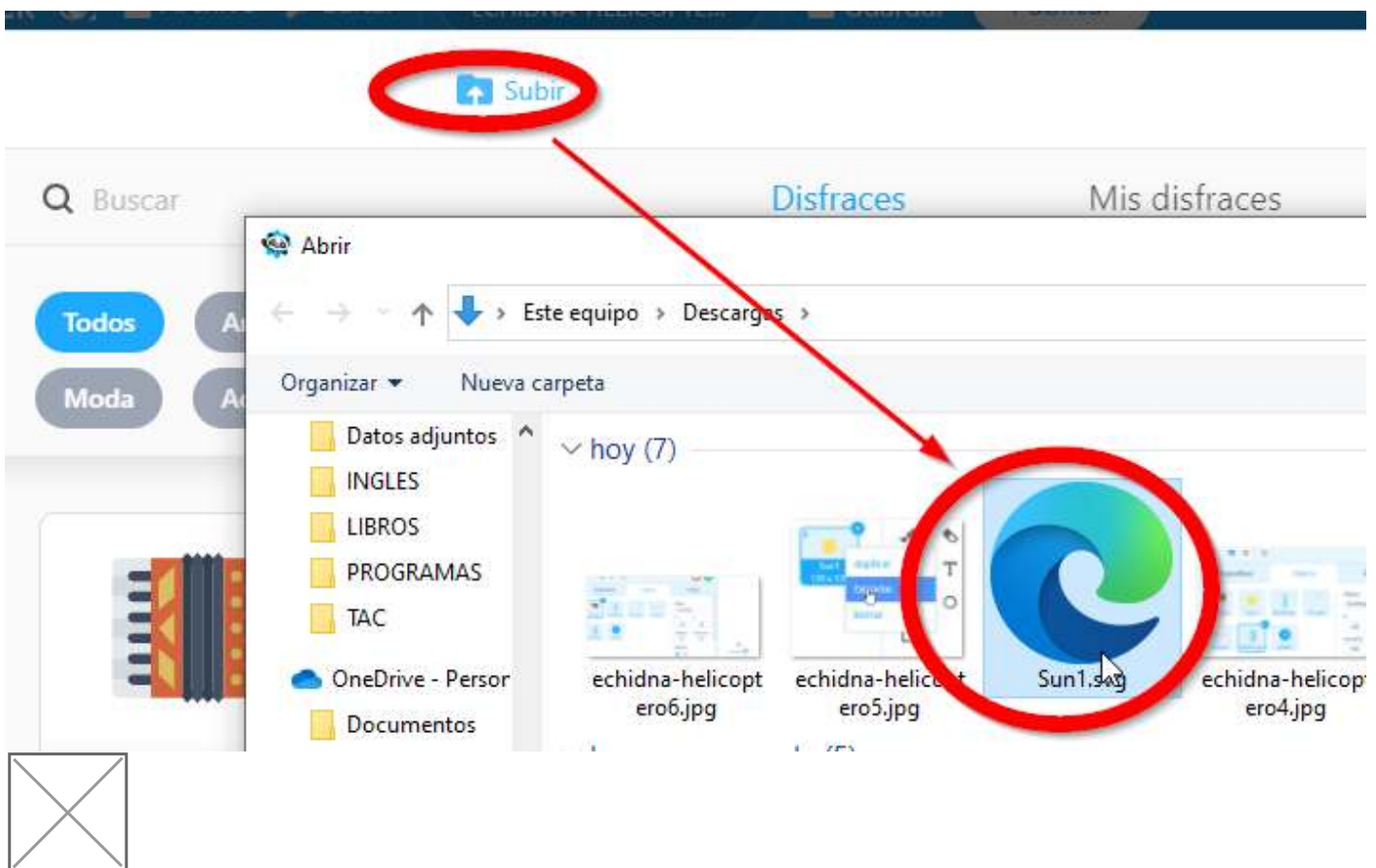
2.- Exporta el fichero imagen svg



3.- Añade al helicóptero el disfraz



4.- Eligiendo el fichero svg exportado en el paso 3



B12 Movimiento helicóptero

Vamos a hacer que helicóptero que se mueva según el acelerómetro: (Ver al final DETALLES MAPEO)



En el arduino le asignamos las coordenadas a las variables globales A2 y A3



B2 Solución edificio, cómo se reutiliza un sprite

B21 Exportación de un Sprites externos

Esta parte es igual que A21 luego no lo repetimos

B22 Importación de Sprites externos

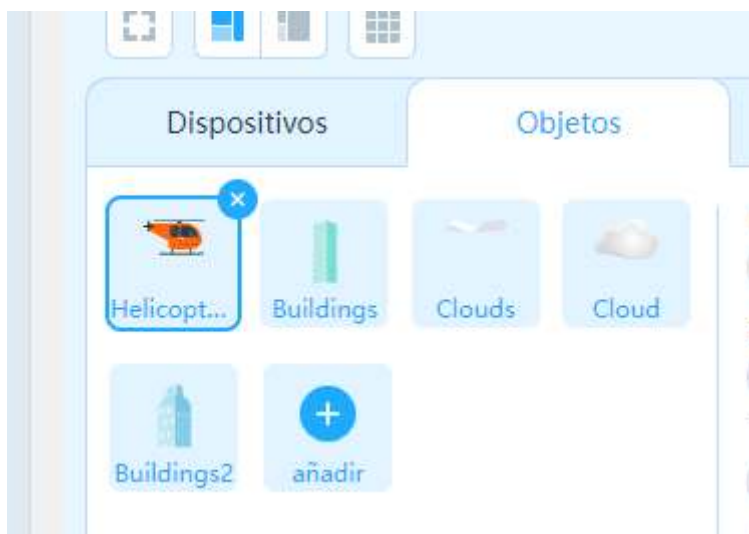


Vamos a mblock e **IMPORTAMOS** el fichero que hemos exportado



Nos importa con sus disfraces Y SU CÓDIGO

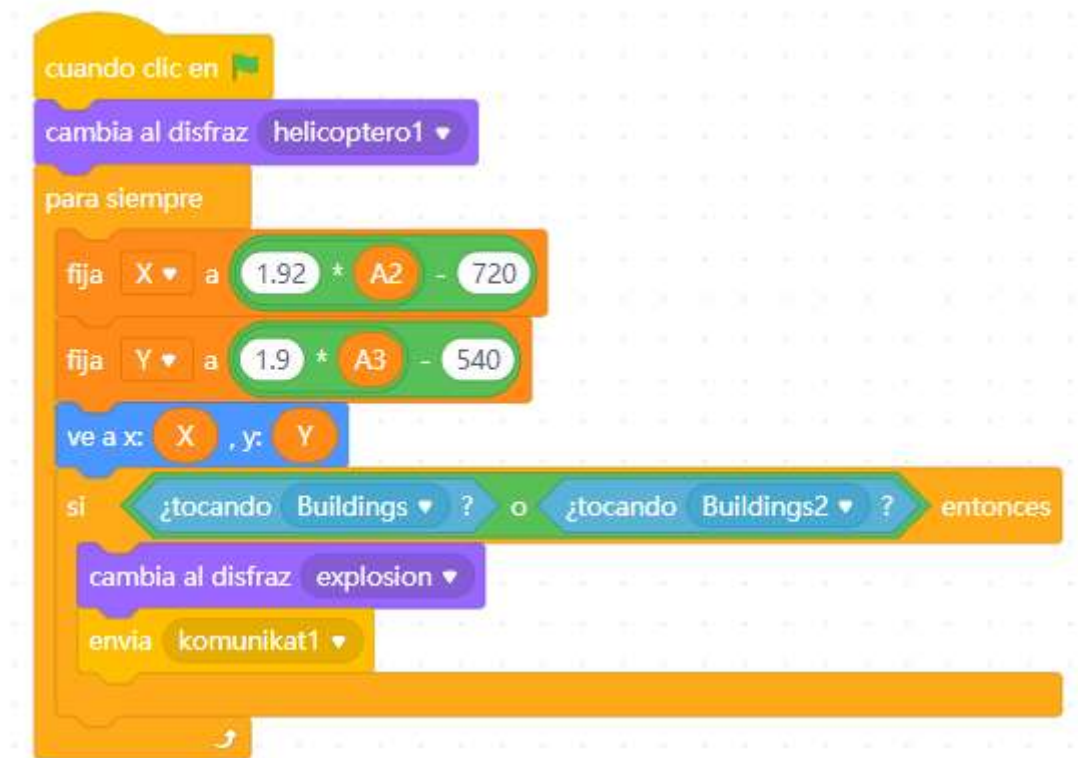
Nos queda pues esta colección de objetos



B3 Solución a la muerte del helicóptero



Vamos a poner que si toca un edificio, “explota” (o sea, cambia el disfraz a explosión), además envía el mensaje “komunikat1” que vemos que el script del edificio lo necesita para esconderse una vez terminado el juego (debe de ser ruso el mensaje es del autor del objeto edificio) :



El programa lo tienes aquí <https://planet.mblock.cc/project/3230430>

MEJORAS

Ya sé que colocando el helicóptero siempre arriba, siempre ganas!!

HAZ QUE SOLO SI TOCA BUILDING1 MUERE, PERO SI TOCA A BUILDING2 QUE GANE UN PUNTO de esta manera ya no se gana poniendo el helicóptero arriba del todo

Detalles MAPEO verlo en la situación E de mapeo:

<https://libros.catedu.es/books/echidna/page/mapeo>

PUEDES CAMBIAR ESTOS VALORES TESTEANDO LOS MOVIMIENTOS DEL SPRITE

MONTAJE 10 MATA-ALIENS hackeando código

Esta vez no vamos a ser originales, vamos a ser un poco vagos y nos vamos a copiar el programa de otra persona. La [web de Scratch](#) tiene un repositorio de millones de proyectos, podemos seleccionar uno, y adaptarlo a nuestra simpática Echidna.



Vamos a ver un ejemplo con este proyecto <https://scratch.mit.edu/projects/29744/>. El reto que te lanzamos es que funcione la nave espacial con nuestro acelerómetro de la Echidna.

IMPORTANDO PROYECTO EXTERNO SCRATCH A EchidnaScratch o mBlock

En [esta presentación](#) se muestra cómo se importa un proyecto a EchidnaScratch o mBlock: (las capturas están con mBlock3, para mBlock5 simplemente no es abrir, sino **Importar desde el ordenador**)

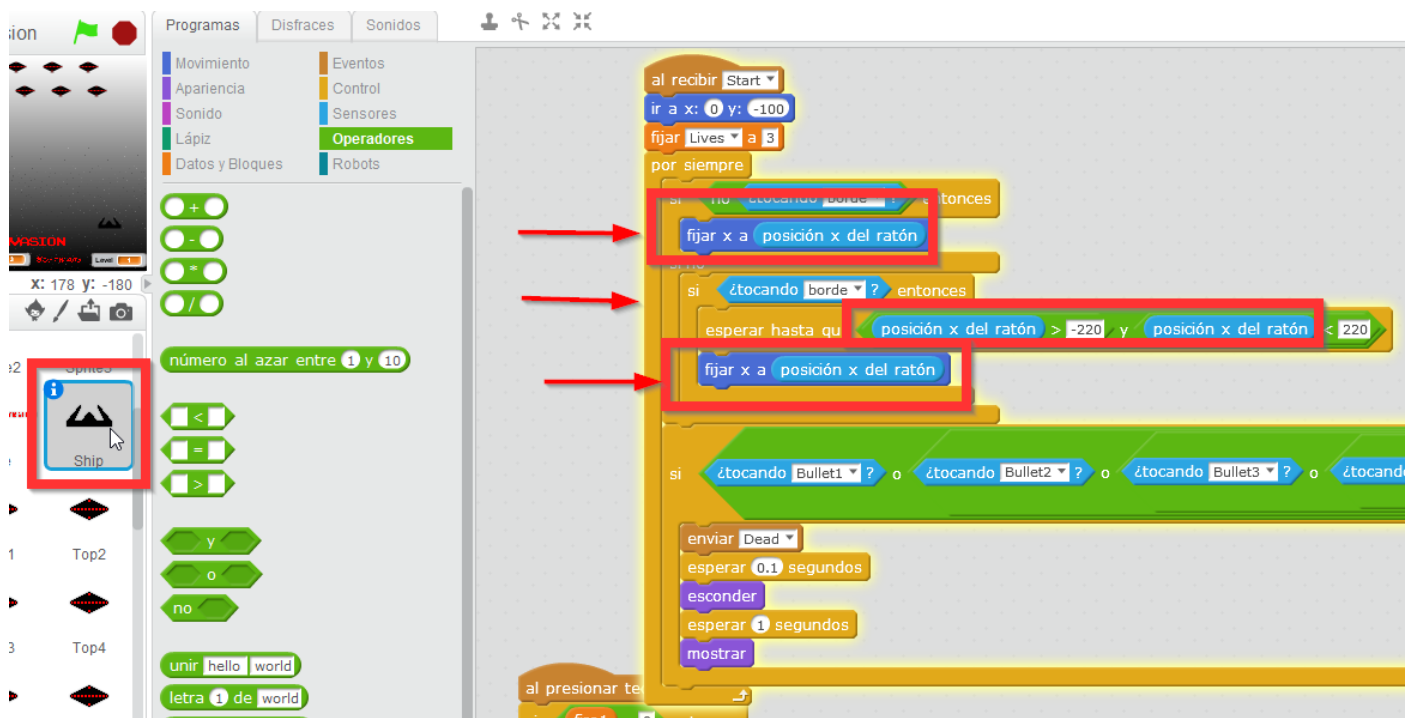
Curiosamente con mBlock3 sale algún fallo pero con EchidnaScratch y mBlock5 no hay, lo importa perfectamente

https://docs.google.com/presentation/d/e/2PACX-1vSCDGNyTkUuXsIOtMJDozS3jijvI8ZgYk9cy50vpgvICAUVZOZbLtTjynKjkZ_CHhq9nvJQzlefO9V2/emb

[ed?start=false&loop=false&delayms=3000](#)

Hackear el código para que funcione con el acelerómetro

Una vez importando nos fijamos en el código que hay que modificar, está claro que lo que hay que tocar es el **Sprite "Ship"** que es el que queremos mover con el acelerómetro y modificar el "posiciónX que se realiza con el ratón y cambiarlo por el acelerómetro:



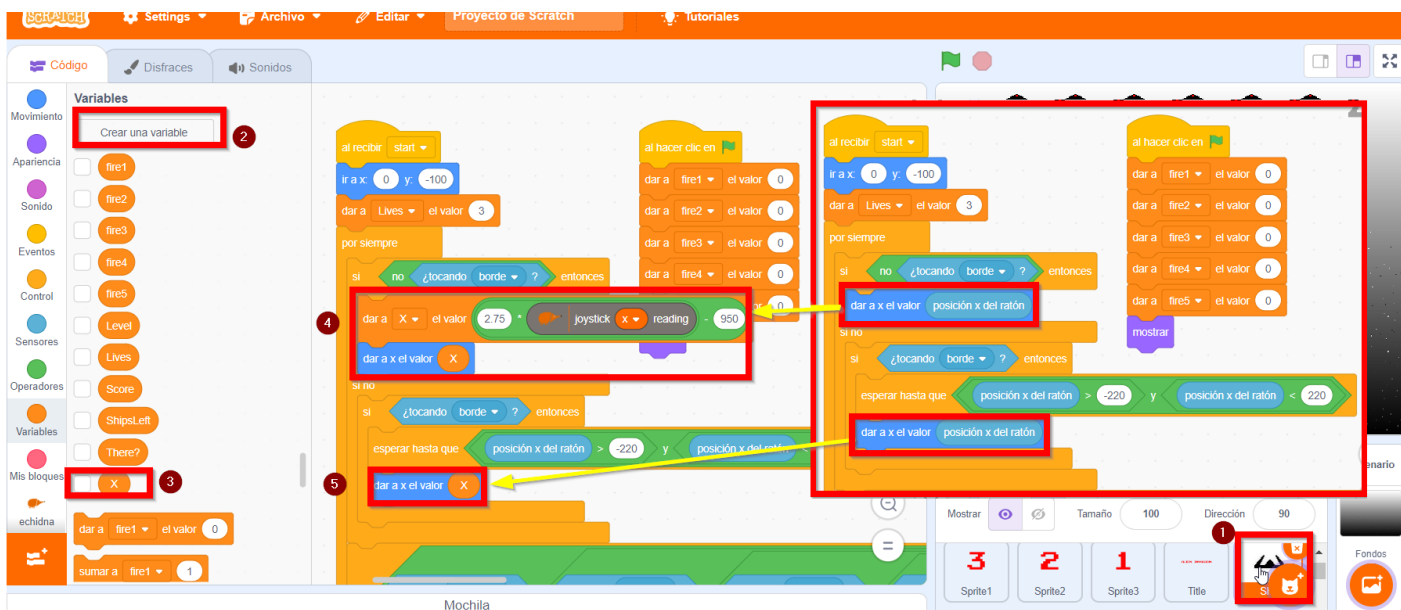
Perooo como siempre: la posición del ratón es la misma que las coordenadas de la pantalla de mBlock y EchidnaScratch y la x va desde -240 hasta 240 y nuestro acelerómetro de A2 va desde 250 hasta 500

Detalles del mapeo en **mapeo situación E**
<https://libros.catedu.es/books/echidna/page/mapeo>

Y cambiamos el código:

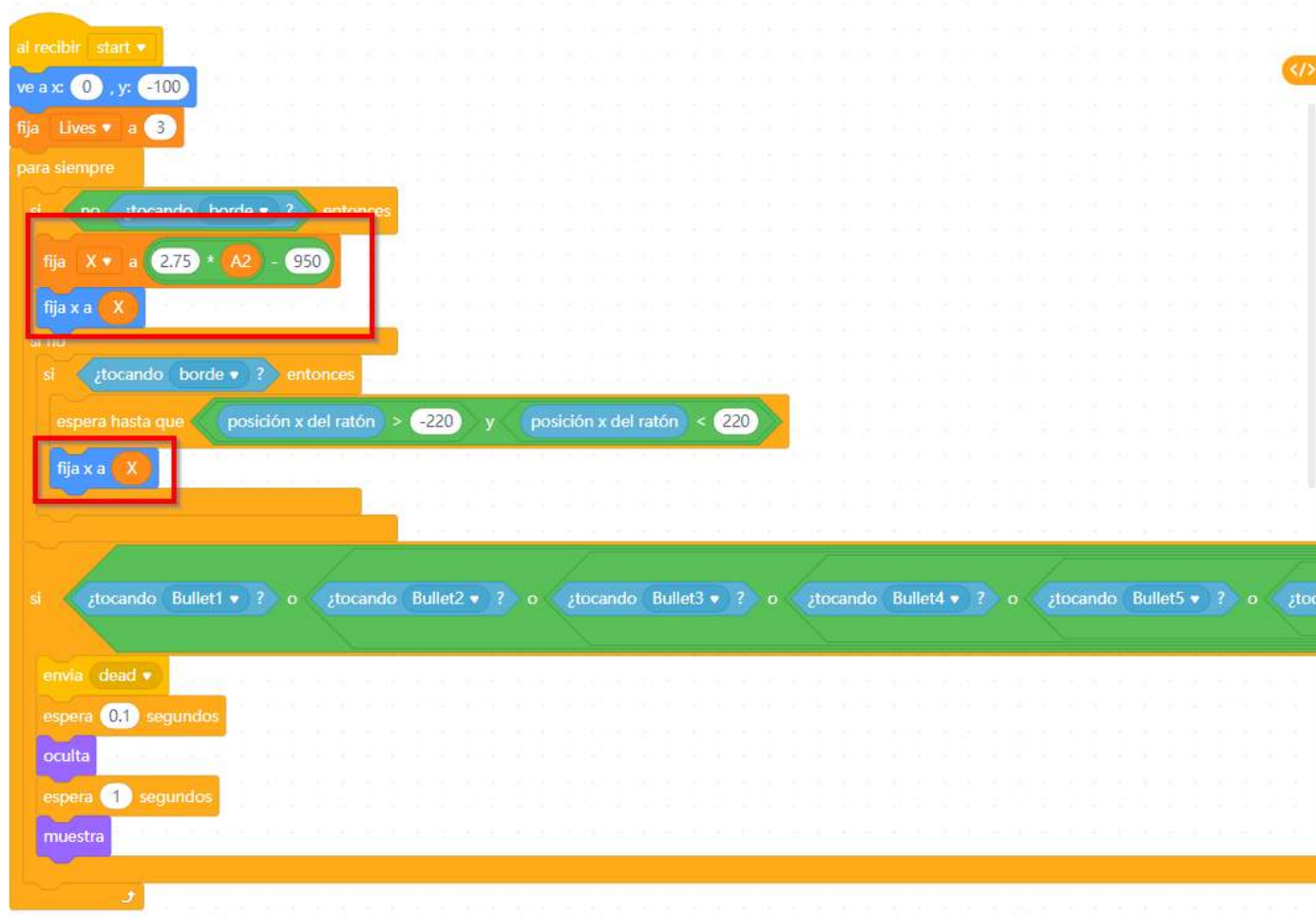
En EchidnaScratch

Cambiamos el código que se recuadra



En mBlock

Idéntico pero como no podemos usar instrucciones específicas de Ecihdna, hay que usar A2:





Donde A2 se tiene que fijar en este código en el Device=dispositivo **Arduino** que tenemos que añadir. Se crean las variables A2 (y también D2 luego lo comentamos)



Un poco más, disparar con D2

También vamos a cambiar el disparador “espacio” :

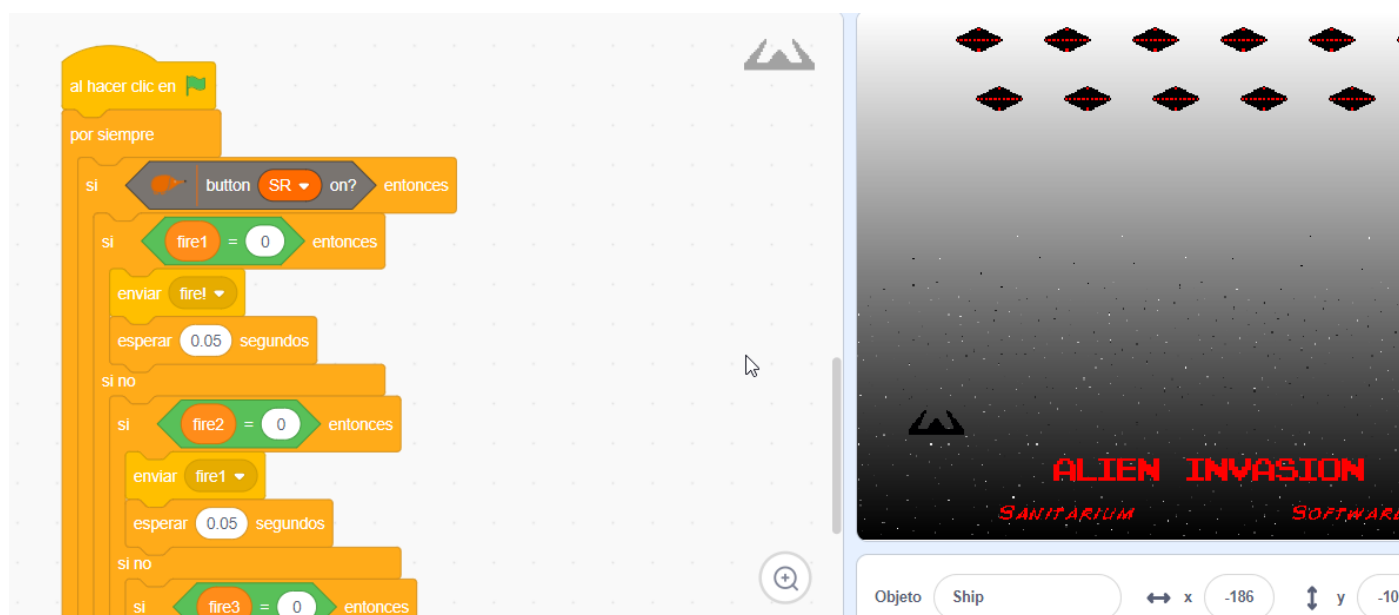


En EchidnaScratch

Quitamos el evento "al presionar la tecla espacio" y lo sustituimos por el evento bandera, con un bucle continuo y una condicional si se aprieta el botón SL de Echidna

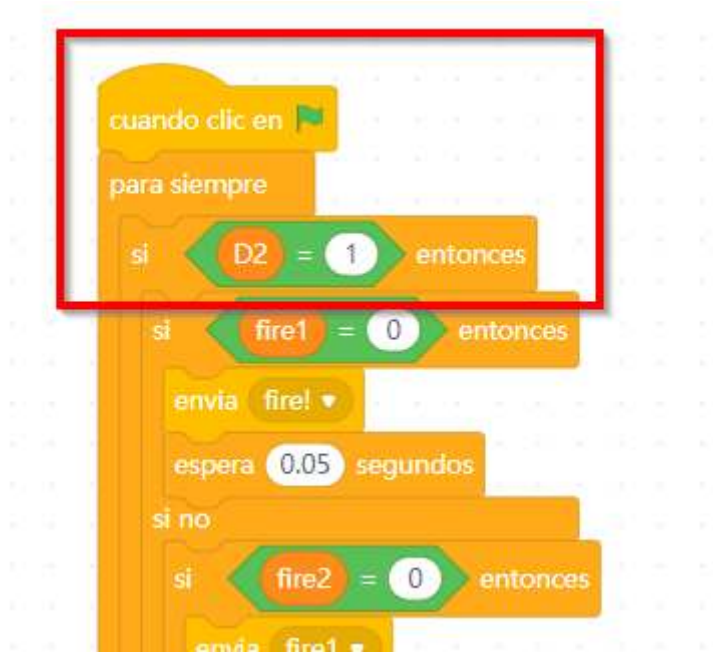


Tiene que quedar así:



En mBlock

No hay instrucción específica de Edhidna, luego usaremos la variable D2 que hemos definido anteriormente que lee el pin D2



Resultado

- En **EchidnaScratch** en este repositorio: <https://github.com/JavierQuintana/Echidna>
- En **mBlock** el programa lo tienes aquí
<https://planet.mblock.cc/project/projectsshare/3232624>

El resultado en este vídeo:

<https://www.youtube.com/embed/Ps3jB89Klg8>

SOLUCIÓN CON SNAP4ARDUINO

Importar un proyecto de Scratch a Snap4Arduino

Para importar un proyecto desde Scratch es un poco más complicado, ver [esta presentación](#):

<https://docs.google.com/presentation/d/e/2PACX-1vQ7PutZ1rJdylUGdaQ6qgkoofiQu1-t0inaBN5JN0idYOlvIK84qa1THTBzNUx4SW3ksIXdjUdEs9NF/embed?start=falseloop=falsedelayms=3000>



Cambios en el código en Snap4Arduino

Los cambios en el código son los mismos que los anteriores.



Recuerda que en [esta presentación](#) se enseña cómo configurar Snap4Arduino para que se comunique con nuestro Arduino y la Echidna.

El programa lo puedes abrir en Snap4Arduino en esta [URL](#)

El Sprite que tienes que conectar con Arduino es la nave esa con cuernos.

El resultado lo puedes ver en este vídeo

https://www.youtube.com/embed/ec4q_5bbBQQ