3. Entradas de Echidna

- Otra vez acuérdate
- MONTAJE 4 ENCENDIDOAPAGADO
- MONTAJE 4-ENCENDIDOAPAGADO-AVANZADO
- MONTAJE 5 LDR
- MONTAJE 5 SEMAFORO SONIDO
- MONTAJE 6 Piano luminoso
- MONTAJE 7 Telesketch
- MONTAJE 7 BIS MATA ALIENS con JOYSTICK
- MONTAJE 7 BIS MATA-ALIENS con acelerómetro

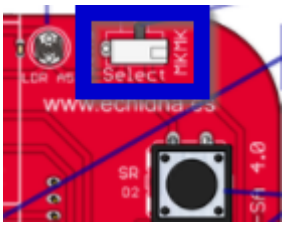
Copyright 2026

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 ENCENDIDOAPAGADO

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.

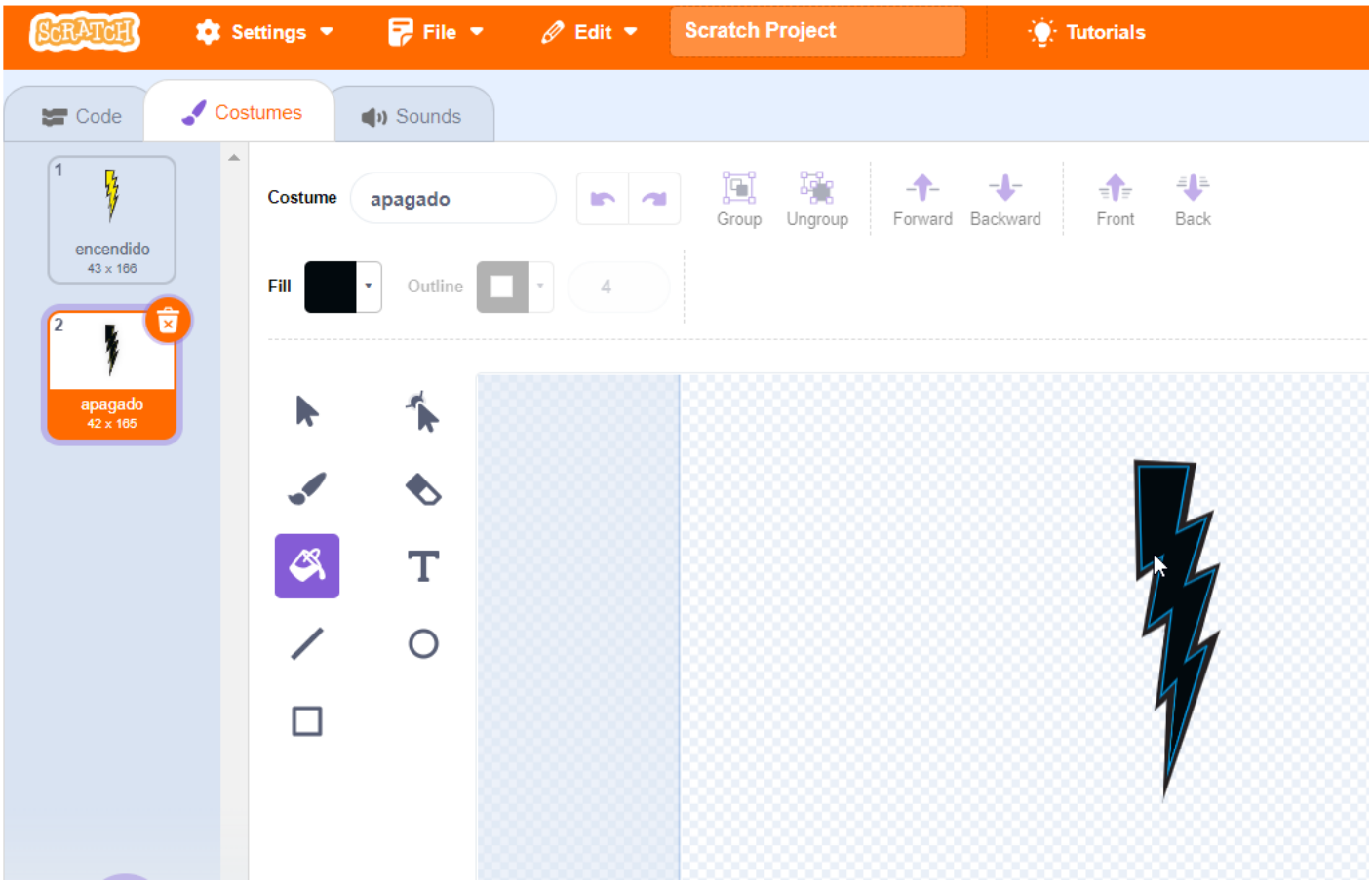
MONTAJE4-ENCENDIDOAPAGADO

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

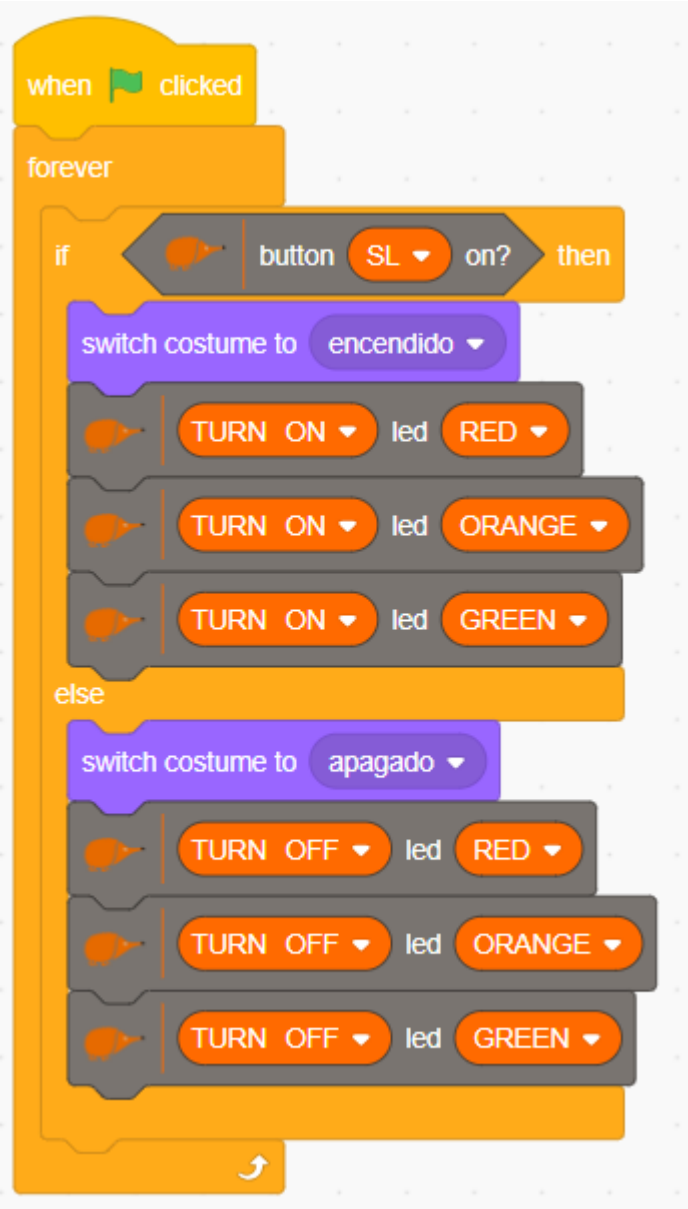
Cambiamos el sprite del gato por otro más acorde y cambiamos el disfraz, no lo desarrollamos aquí pues vimos cómo se hace con el MONTAJE1

Copyright 2026



El código :

Copyright 2026



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

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

Copyright 2026

MONTAJE 4-ENCENDIDOAPAGADO-AVANZADO

MONTAJE4-ENCENDIDOAPAGADO-AVANZADO

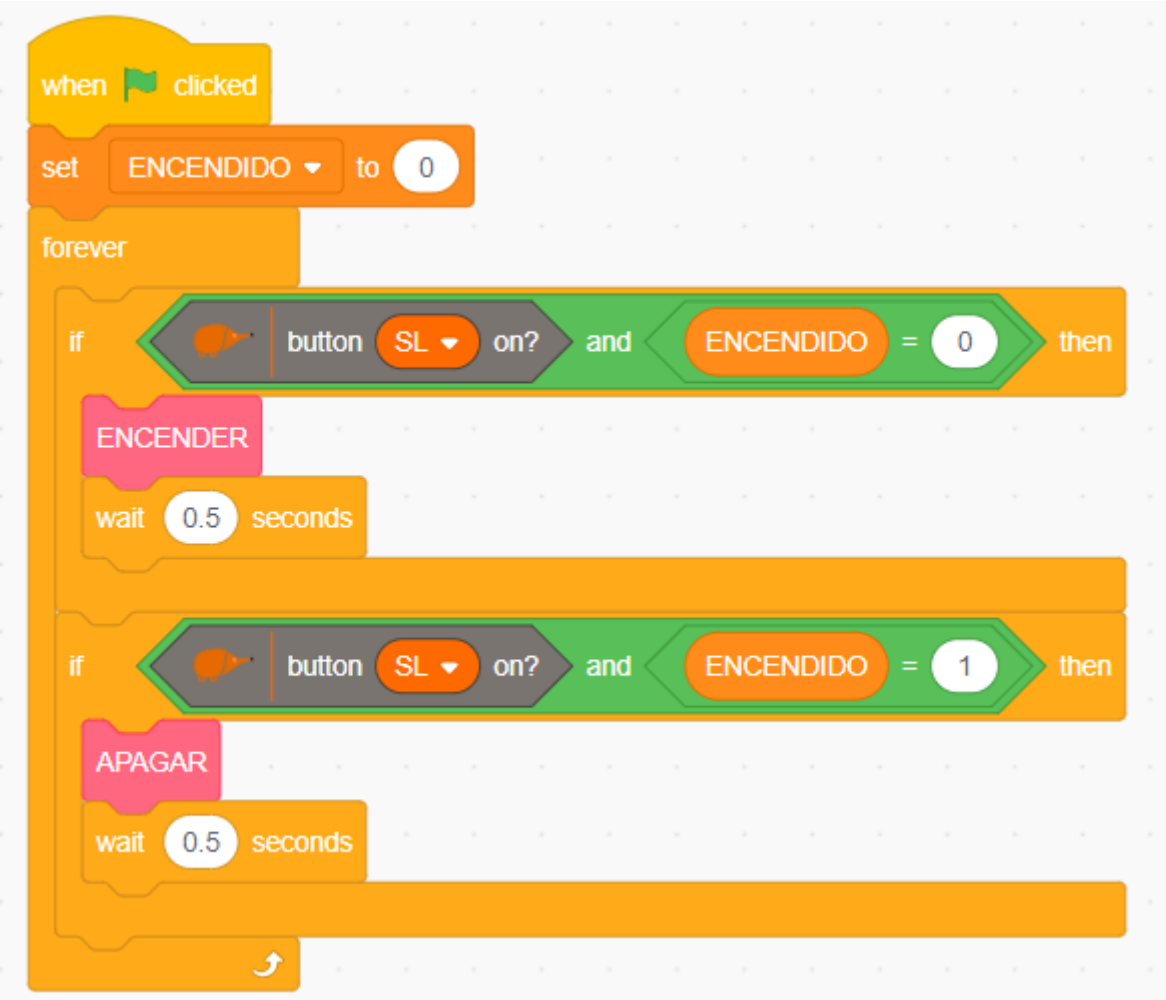
Al pulsar D2 tiene que encenderse los leds, y **sólo se apagaran** si se vuelve a pulsar D2. O sea, utilizar sólo un pulsador para encender y apagar los leds.

Solución EchidnaShield

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

El código es

Copyright 2026



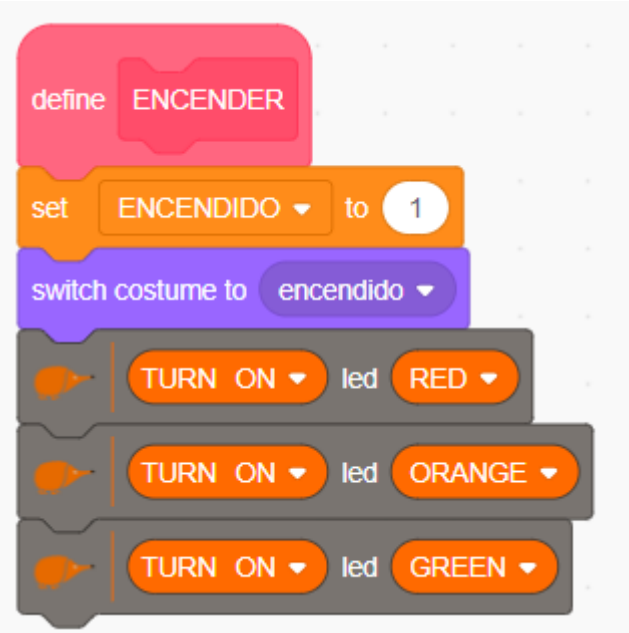
Para evitar que el programa sea engorroso, se ha creado los bloques ENCENDER y APAGAR, pero perfectamente se puede meter estas instrucciones dentro del algoritmo principal y pasar de bloques.

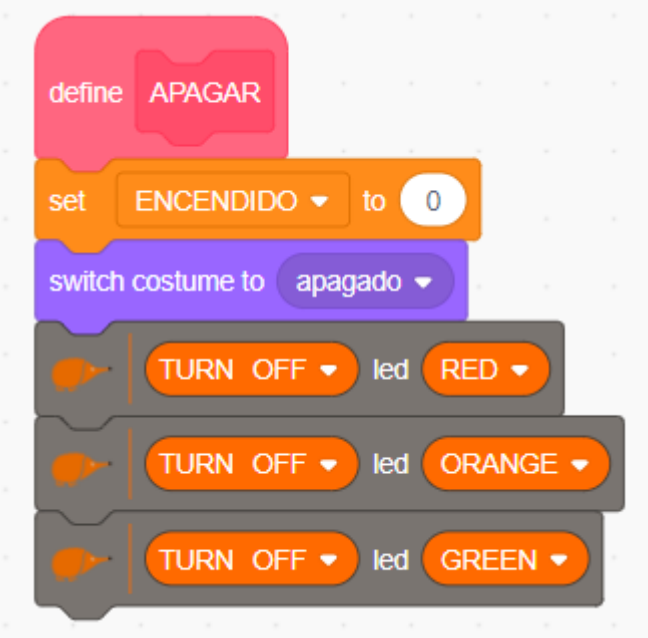
> **FIJATE** □□□ que la variable ENCENDIDO se cambia a 1 o a 0 para que refleje el estado en el que ha quedado el sistema

> **FIJATE** □□ que hay una espera de 0.5segundos para que en ese tiempo **no lea** si no lo pones, esta todo el rato cambiando los pocos milisegundos que pulsas el botón
> Una manera de quitar ese efecto es poner un "Esperar que NO este pulsado" mira:



Manual EchidnaBlack[2] y EchidnaML www.echidna.es CC-BY-SA





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

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

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.

Copyright 2026

MONTAJE 5 LDR

¿Qué es un LDR?

Vamos a la Wikipedia:



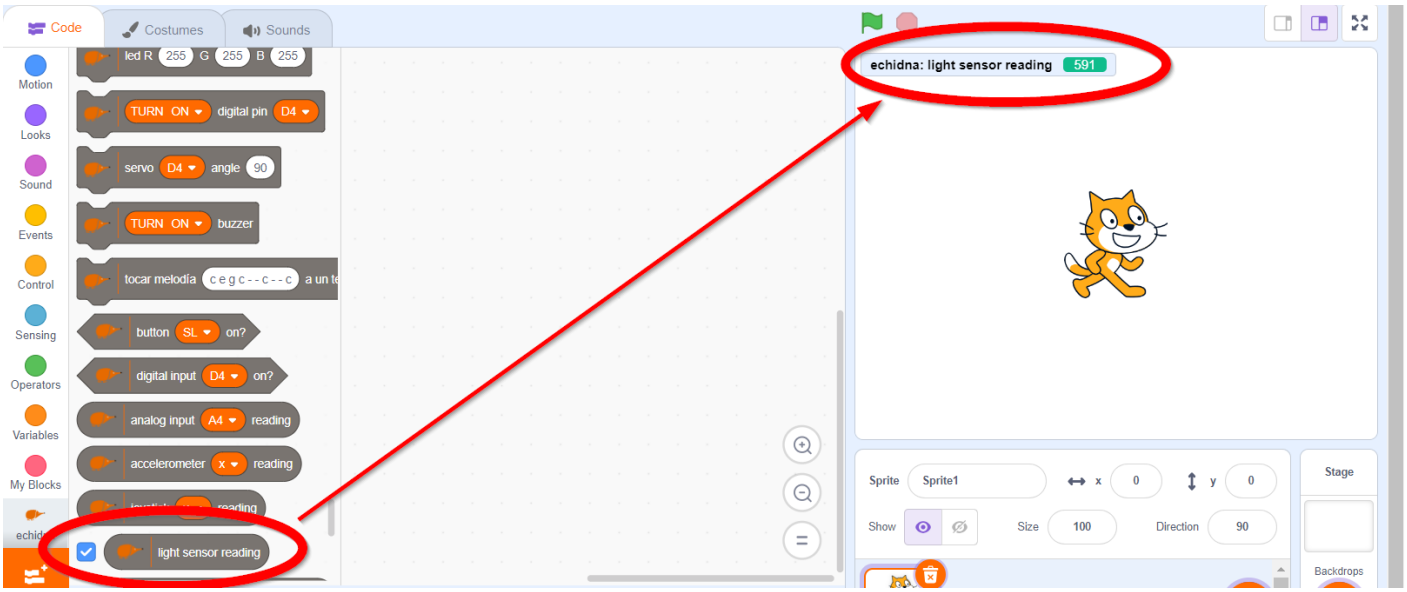
> Fotorresistor *Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado fotorresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuyas siglas, LDR, se originan de su nombre en inglés 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

Copyright 2026



¿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

Copyright 2026

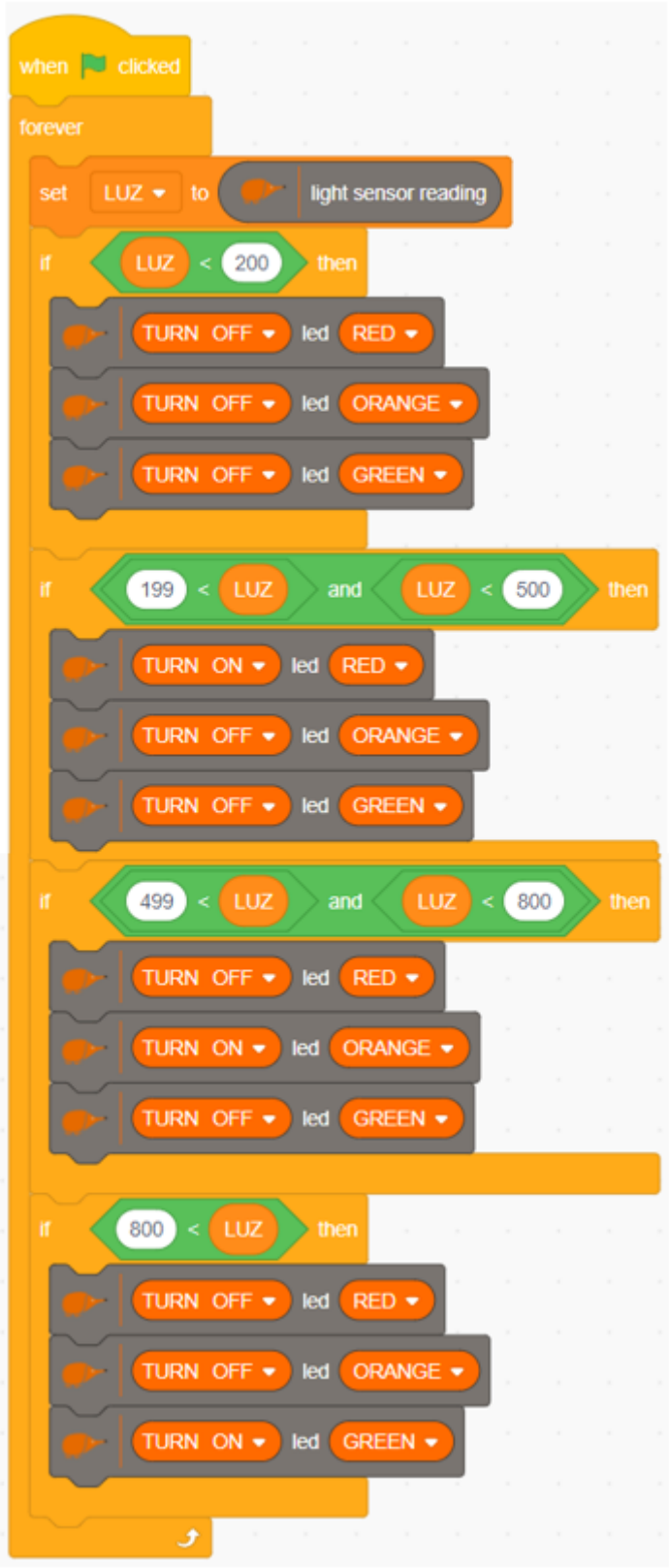
Semáforo	Luz
Amarillo	Luz normal
Verde	Mucha luz

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

Copyright 2026

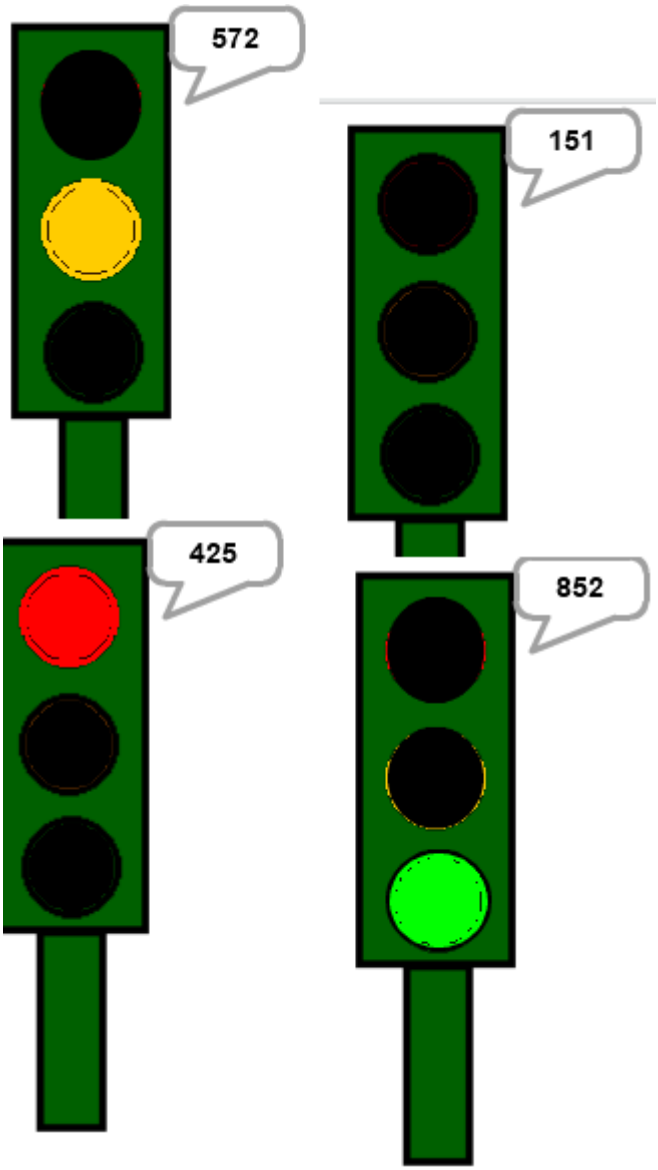


https://www.youtube.com/embed/5JJY39A2Fm8

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

Copyright 2026

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

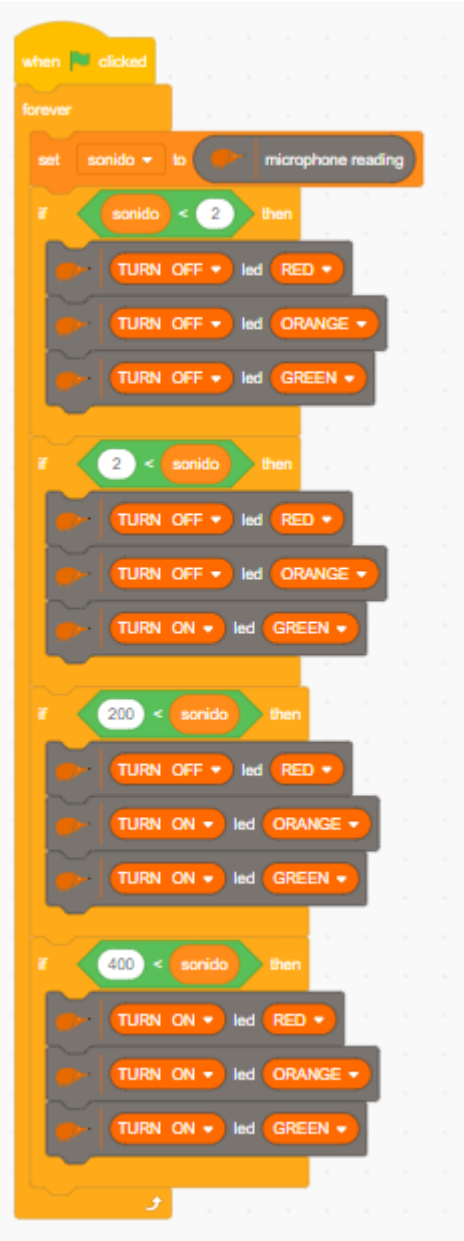
> Vamos a hacer un semáforo que mida la intensidad del sonido, verde si es poco, y sube hasta rojo cuando es elevado

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

https://github.com/JavierQuintana/Echidna

El programa en EchidnaScratch es :

Copyright 2026



Hemos simulado el sonido soplando en el micro :

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

> **¿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)

Copyright 2026

MONTAJE 6 Piano luminoso

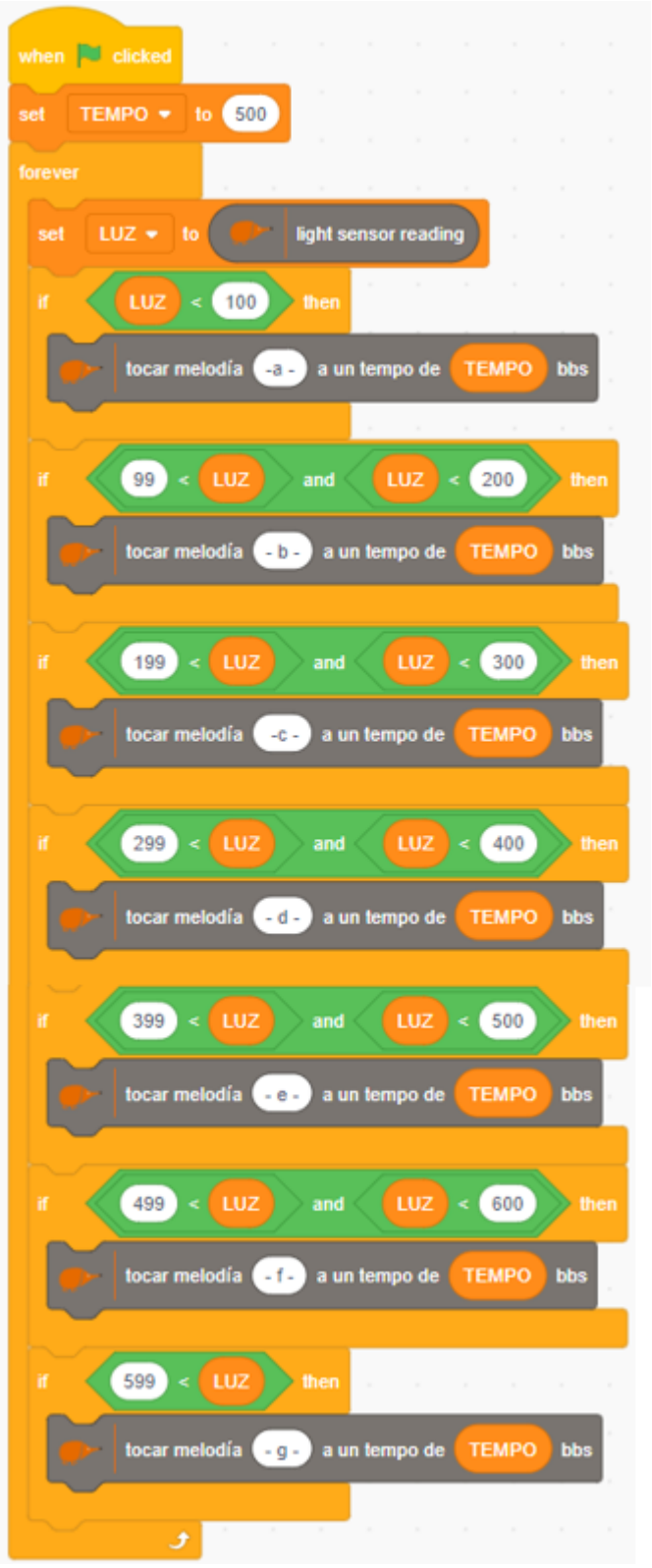
Enunciado

> *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

Copyright 2026



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

Copyright 2026

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

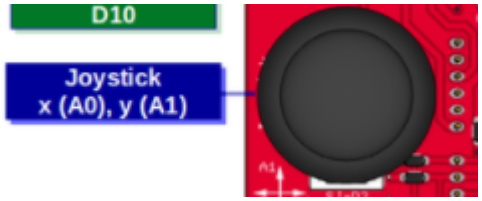
https://github.com/JavierQuintana/Echidna

MONTAJE 7 Telesketch

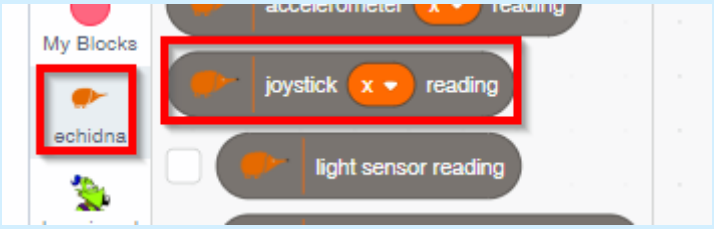
¿Qué es el Joystick?

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

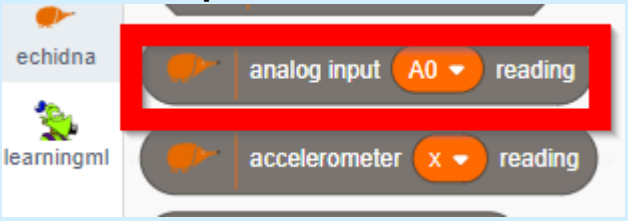
El potenciómetro X está conectado en los pines A0 del Arduino y el del eje Y al A1 por lo tanto si has leído los capítulos de ARDUINO **sus valores varía desde 100 hasta 1024.** El pulsador está conectado al D10.



LAS COSAS CLARAS: IGUAL DA ESTA INSTRUCCIÓN



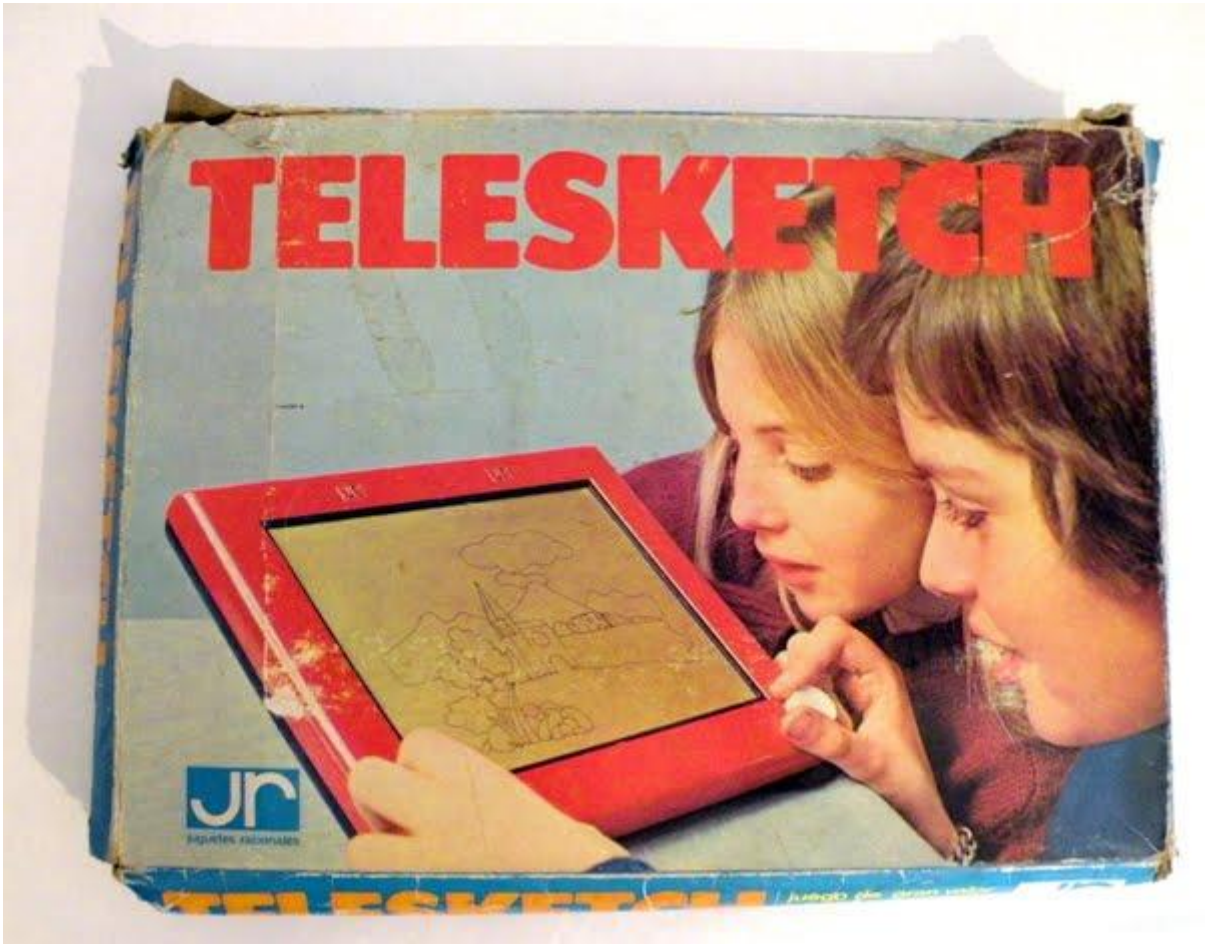
QUE ESTA: (poniendo A0)



Preferimos la PRIMERA pues EchidnaML se adapta a la versión de la placa *es decir, que si en el futuro el joystick x en una versión furura es por ejemplo A5 la instrucción primera se adapta y lo envía a la A5 siendo transparente al alumnado, mientras que en la segunda tienes que saber que ha cambiado a A5*

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

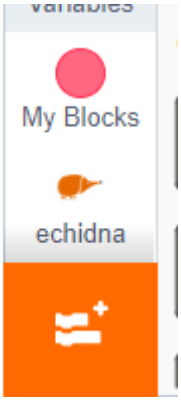
La solución la puedes descargar aquí
https://github.com/JavierQuintana/Echidna/blob/master/MONTAJE7-ECHIDNABLACK2-JOYSTICK-TELESKETCH.sb3

Simplemente vamos a considerar estas condiciones:

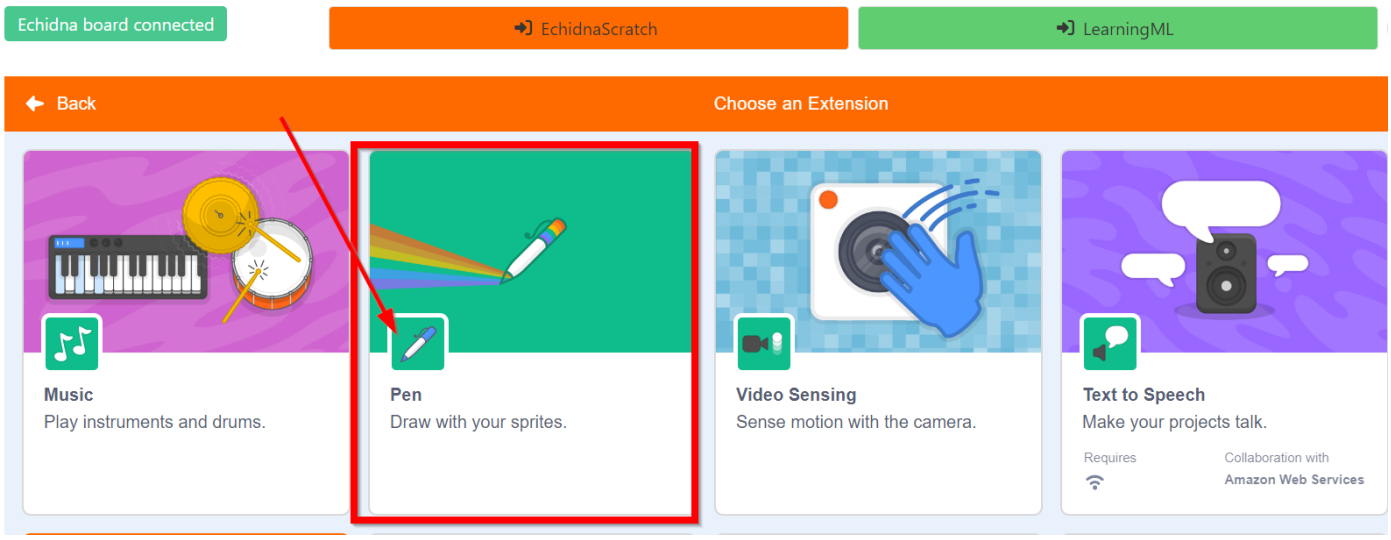
*Si el joystick en el **eje X** es menor que **300** es que estás inclinando el Joystick hacia la izquierda, luego el Sprite tiene que modificar su variable **decrementando** su valor, un valor fijo que lo definiremos como PASO.*
*- Si el joystick en el **eje X** es mayor que **700** es seguro que estás 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?

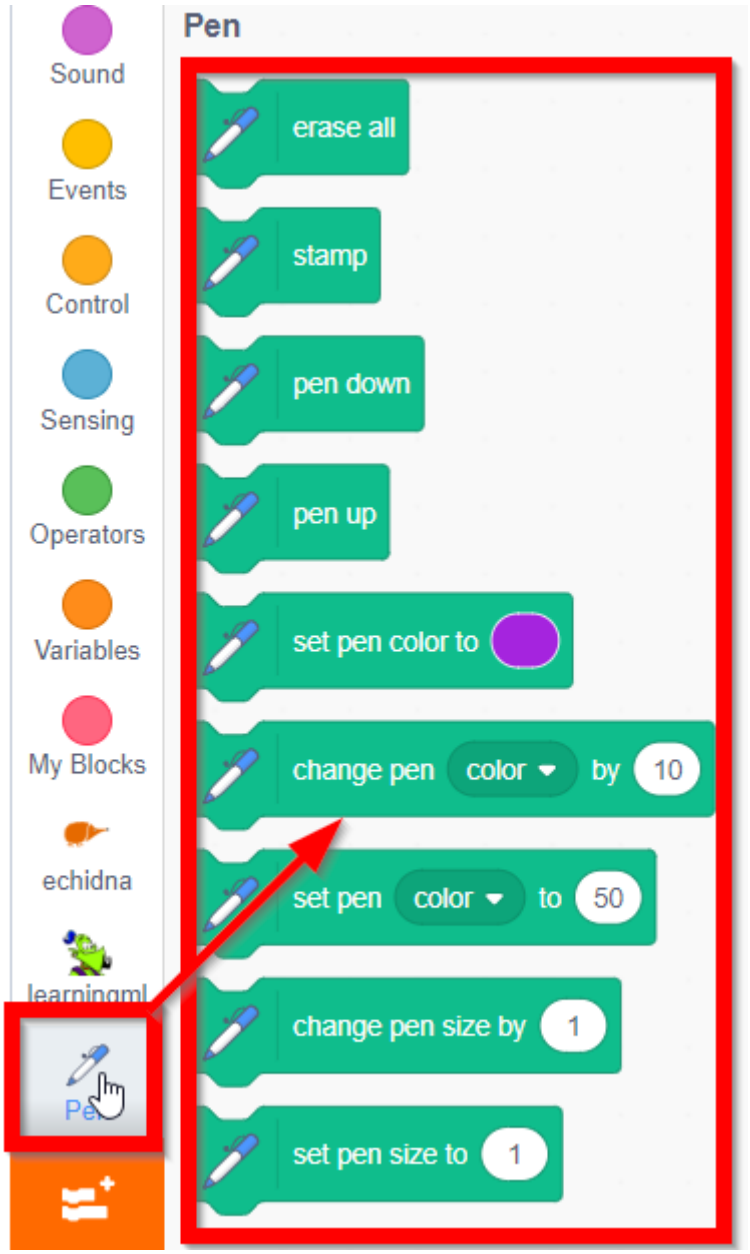
Lápiz es una extensión para poder dibujar. Para instalar una extensión pulsa aquí



E instalamos esta extensión

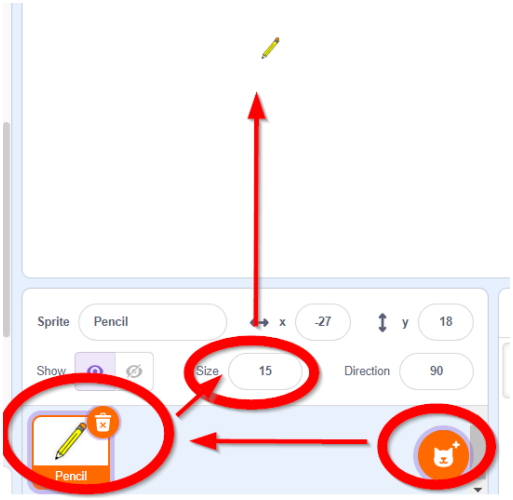


Y se instalan nuevas instrucciones para dibujar



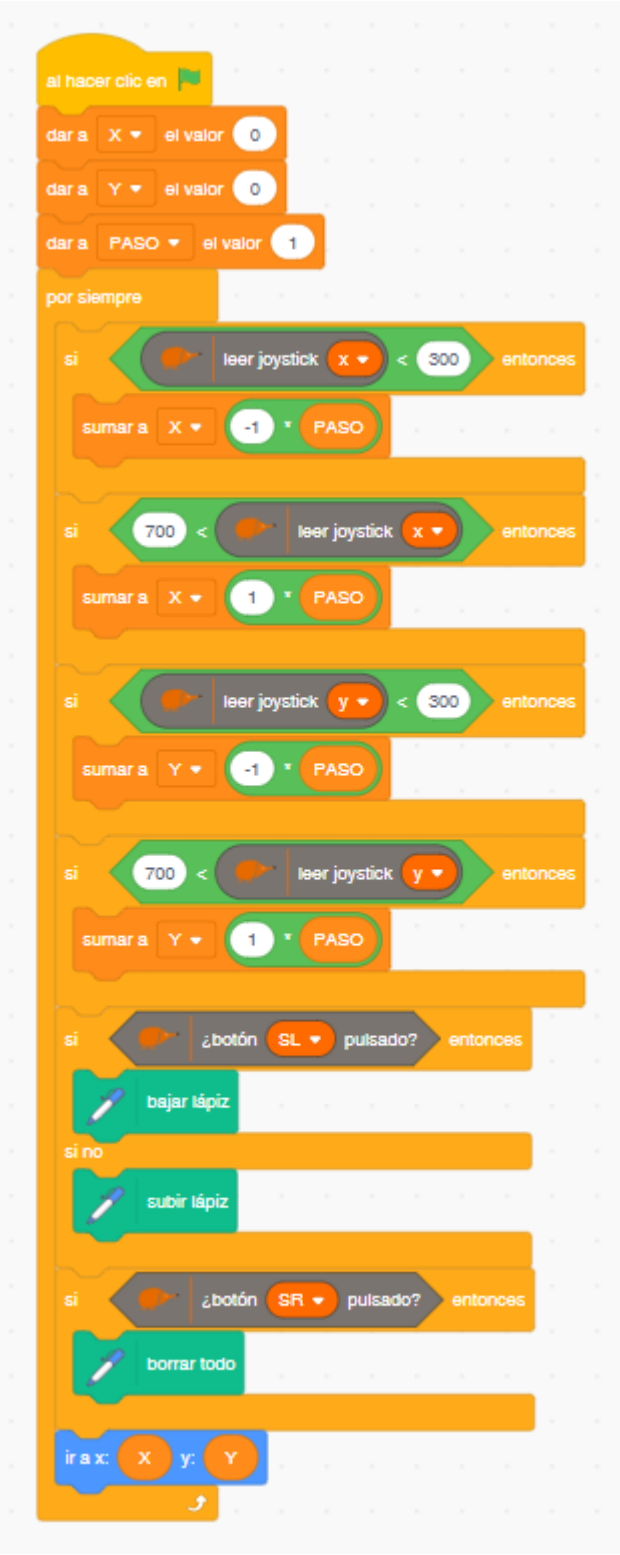
Código con EchidnaScratch

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



Te lo puedes descargar aquí https://github.com/JavierQuintana/Echidna/blob/master/MONTAJE7-ECHIDNABLACK2-JOYSTICK-TELESKETCH.sb3

Copyright 2026

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

Copyright 2026

MONTAJE 7 BIS MATA ALIENS con JOYSTICK

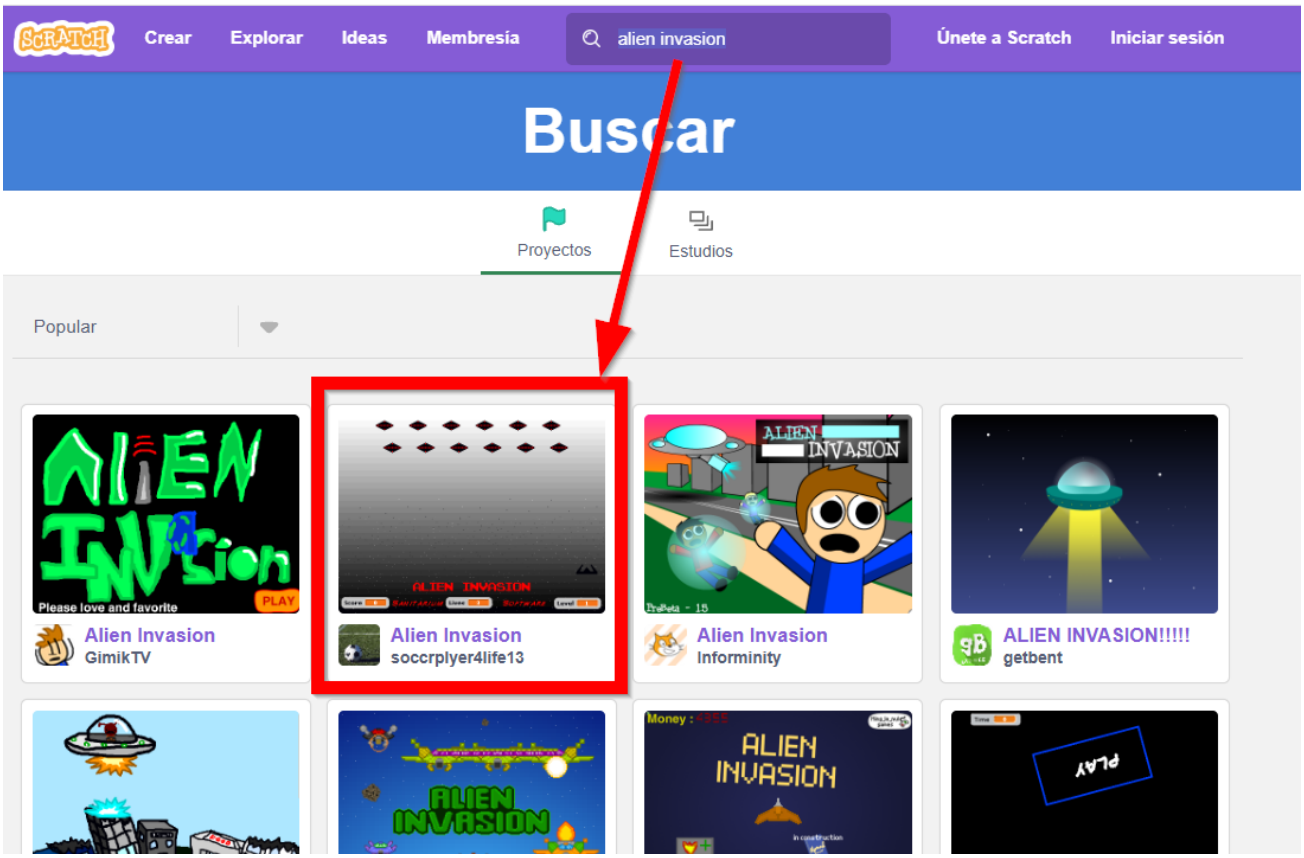
Esta vez no vamos a ser originales, vamos a ser un poco vagos y nos vamos a copiar el programa de otra persona. La web de Scratch tiene un repositorio de millones de proyectos, podemos seleccionar uno, y adaptarlo a nuestra simpática Echidna.



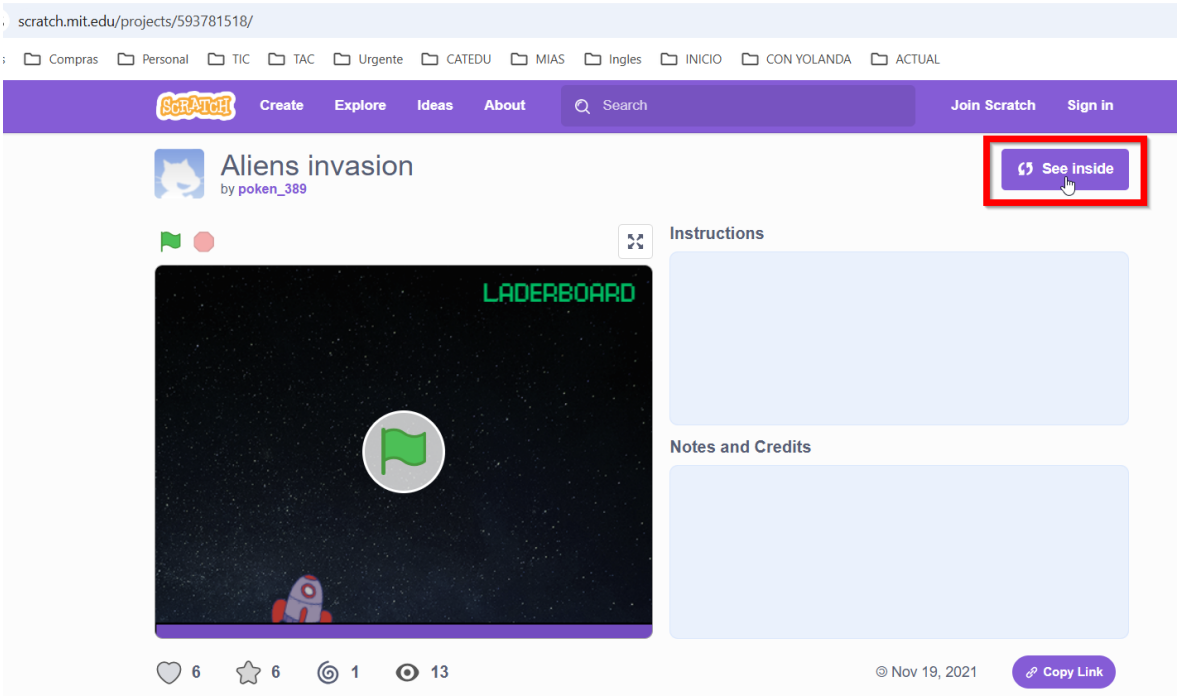
> Vamos a copiar un proyecto ALIENS INVASION nave espacial con nuestro JOYSTICK de la Echidna.

1.- IMPORTANDO PROYECTO EXTERNO SCRATCH A EchidnaScratch o mBlock

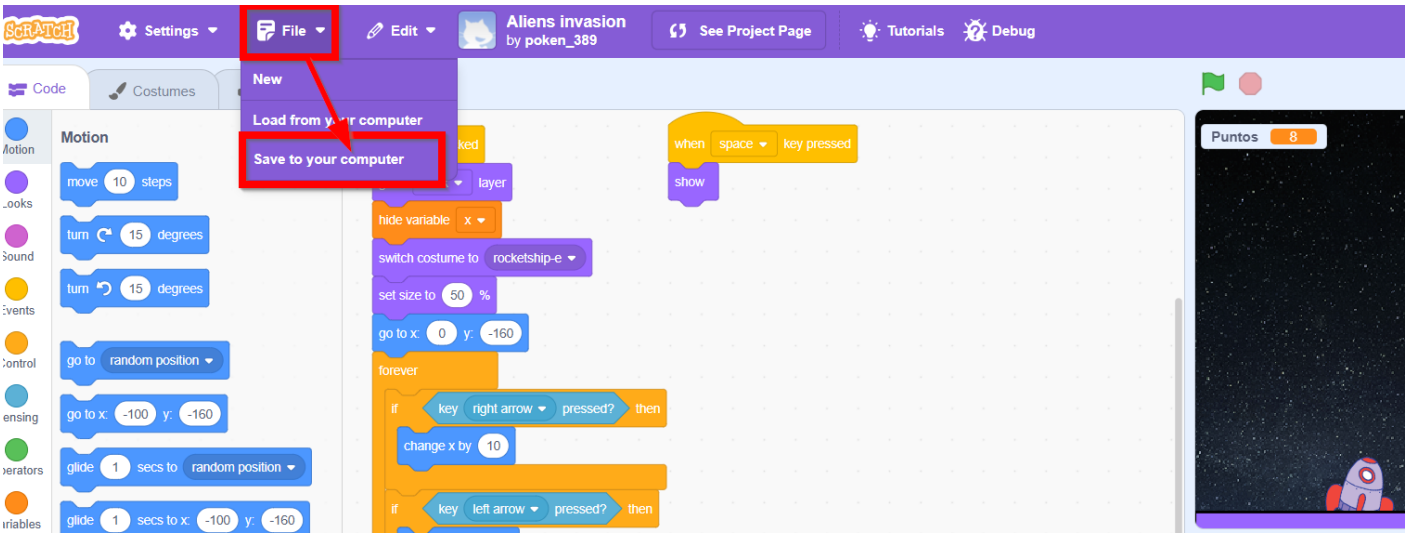
Entramos en SCRATCH https://scratch.mit.edu/ y buscamos ALIEN INVASION, concretamente este : https://scratch.mit.edu/projects/29744/



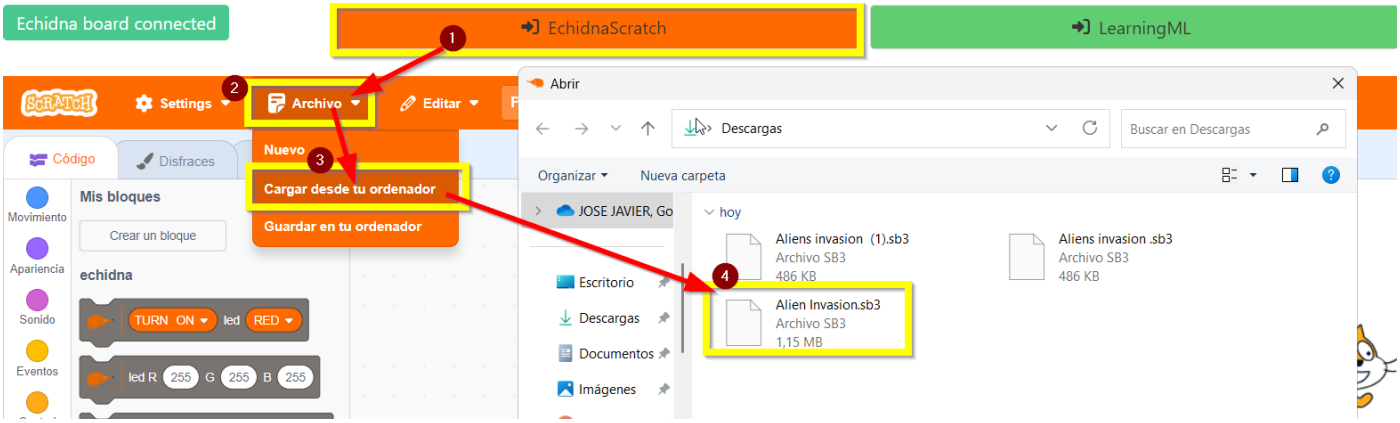
Entramos en **SEE INSIDE**



Lo grabamos en nuestro ordenador

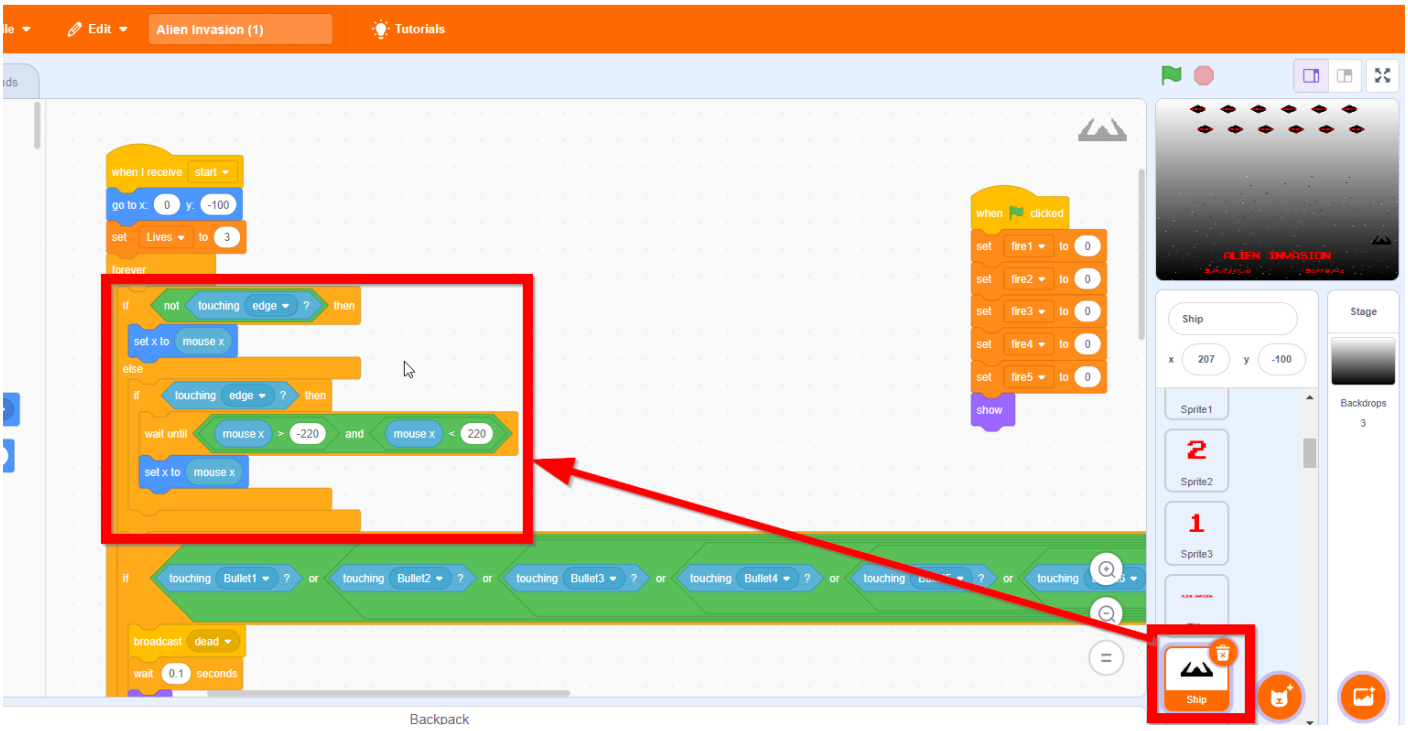


Nos vamos al programa ECHIDNA ML y cargamos el programa descargado



2.- Hackear el código para que funcione la nave con el joystick

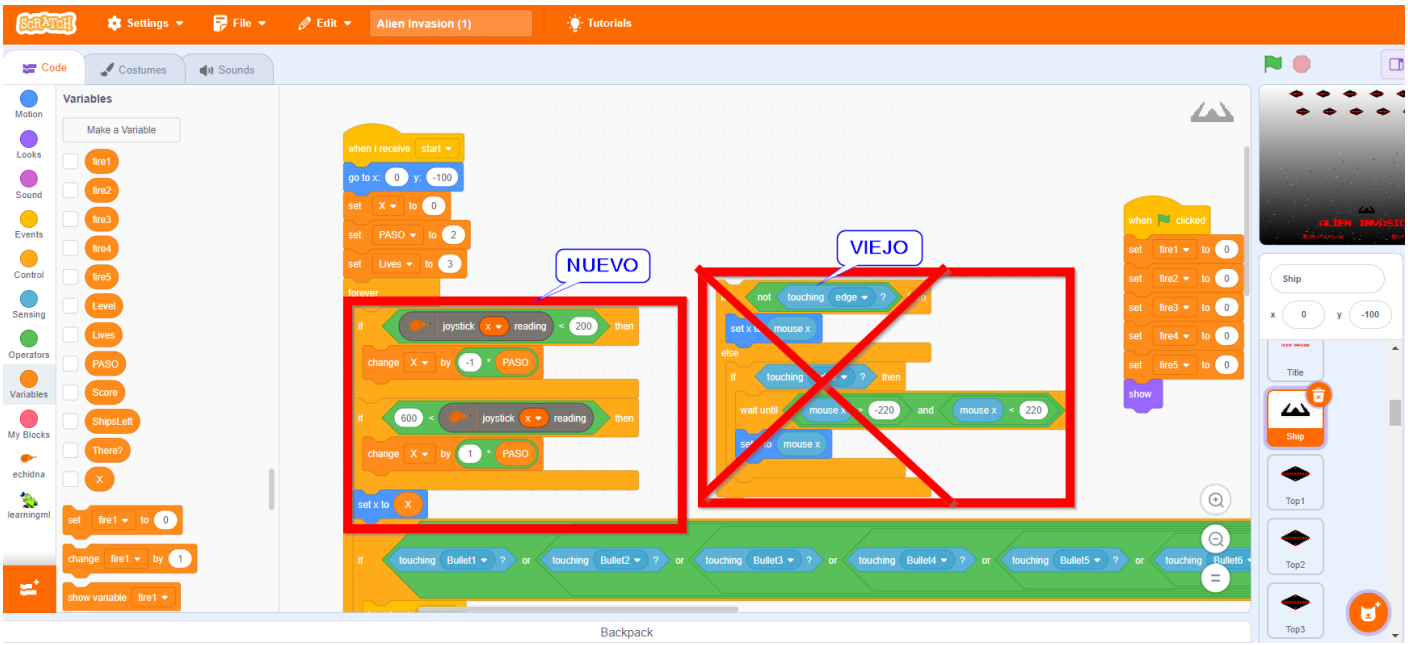
Vamos a fijarnos en el código de la nave, si nos fijamos se mueve según la posición X del ratón, **vamos a cambiar ese código**



Creamos dos variables :

- Una que se llame **X** que tendrá la posición X de la nave, la inicializamos a 0 al principio y así se coloca la nave enmedio al empezar la partida
- Otra que se llame **PASO** que fija el paso o incremento cuando se mueva, cuanto más alto, más rápido se mueve, pero menos preciso, a golpes, cuanto más bajo, más lento pero más preciso, lo pondremos a 2

Y cambiamos el código de mover la nave :

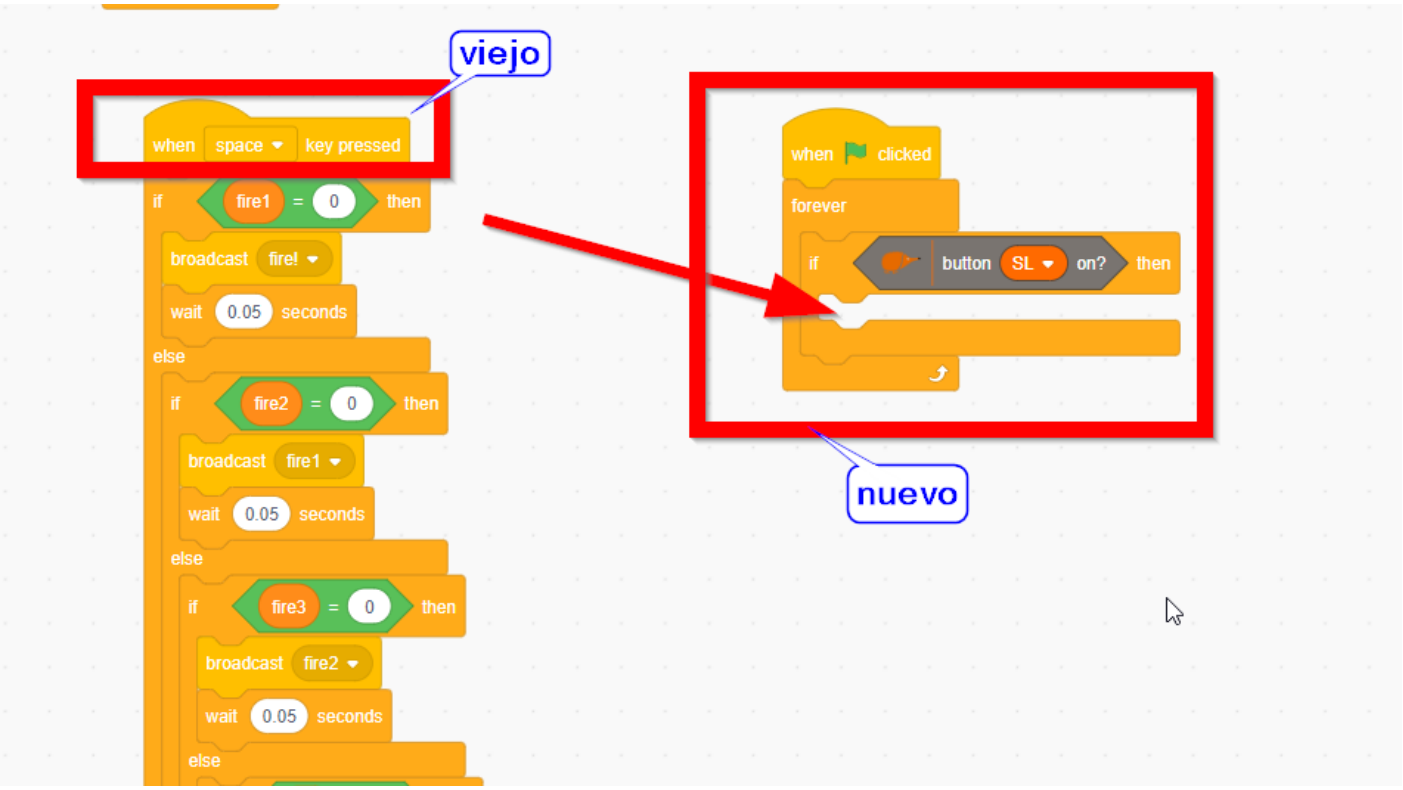


- Si el joystick en el sentido X **baja** de 200 (puede ser un número entre 0-512 pues 512 en teoría es enmedio) pues que X se **decremente** un PASO
- Si el joystick en el sentido X **sube** de 600 (puede ser un número entre 512-1021 pues 1024 es el máximo) pues que X se **incremente** un PASO

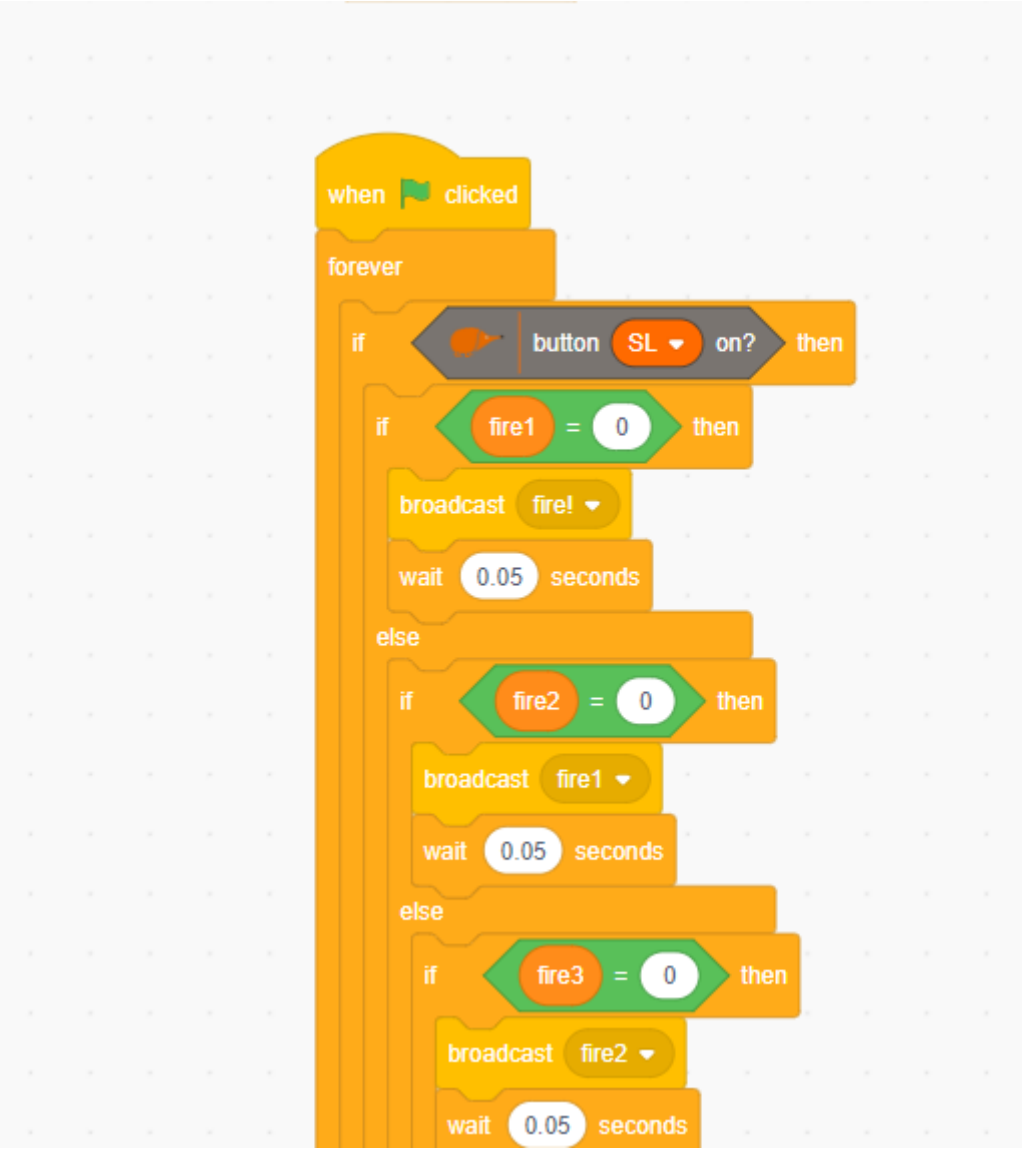
3.- QUE DISPARE APRETANDO EL BOTÓN SL

Si nos fijamos, dispara si pulsamos la tecla **espacio**, lo quitamos y ponemos todo ese código dentro del código nuevo que se ejecuta si apretamos el botón SL

Copyright 2026


viejo
when space key pressed
if fire1 = 0 then
broadcast fire!
wait 0.05 seconds
else
if fire2 = 0 then
broadcast fire1
wait 0.05 seconds
else
if fire3 = 0 then
broadcast fire2
wait 0.05 seconds
else
when clicked
forever
if button SL on? then
nuevo

Queda pues



El resultado

El programa te lo puedes descargar aquí

https://github.com/JavierQuintana/Echidna/blob/master/MONTAJE%207-BIS%20ECHIDNABLACK2%20MATA-ALIENS%20CON%20JOYSTICK%20SIN%20MAPEAR.sb3

https://www.youtube.com/embed/Ps3jB89KIg8

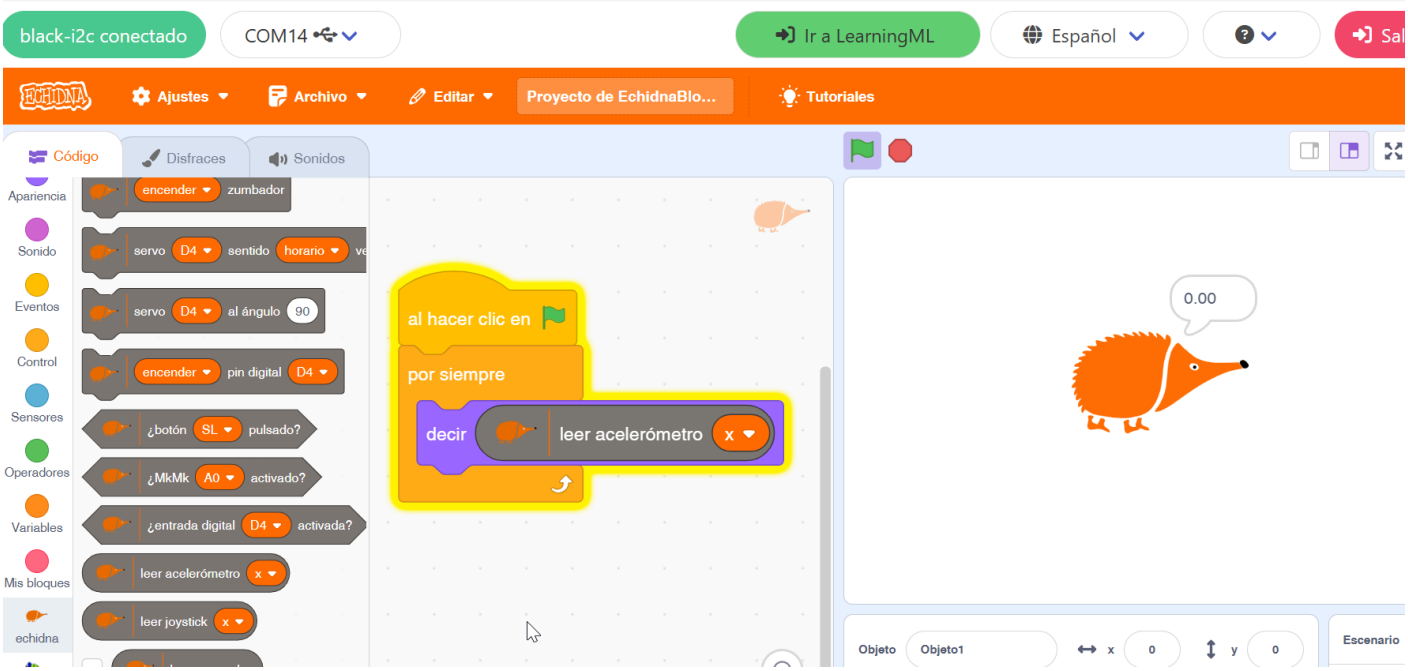
MONTAJE 7 BIS MATA-ALIENS con acelerómetro

¿Se puede hacer el anterior ejercicio con el acelerómetro? Por supuesto que sí, la única diferencia es que el Joystick va de 0 a 1024 pero el acelerómetro tiene otros valores :

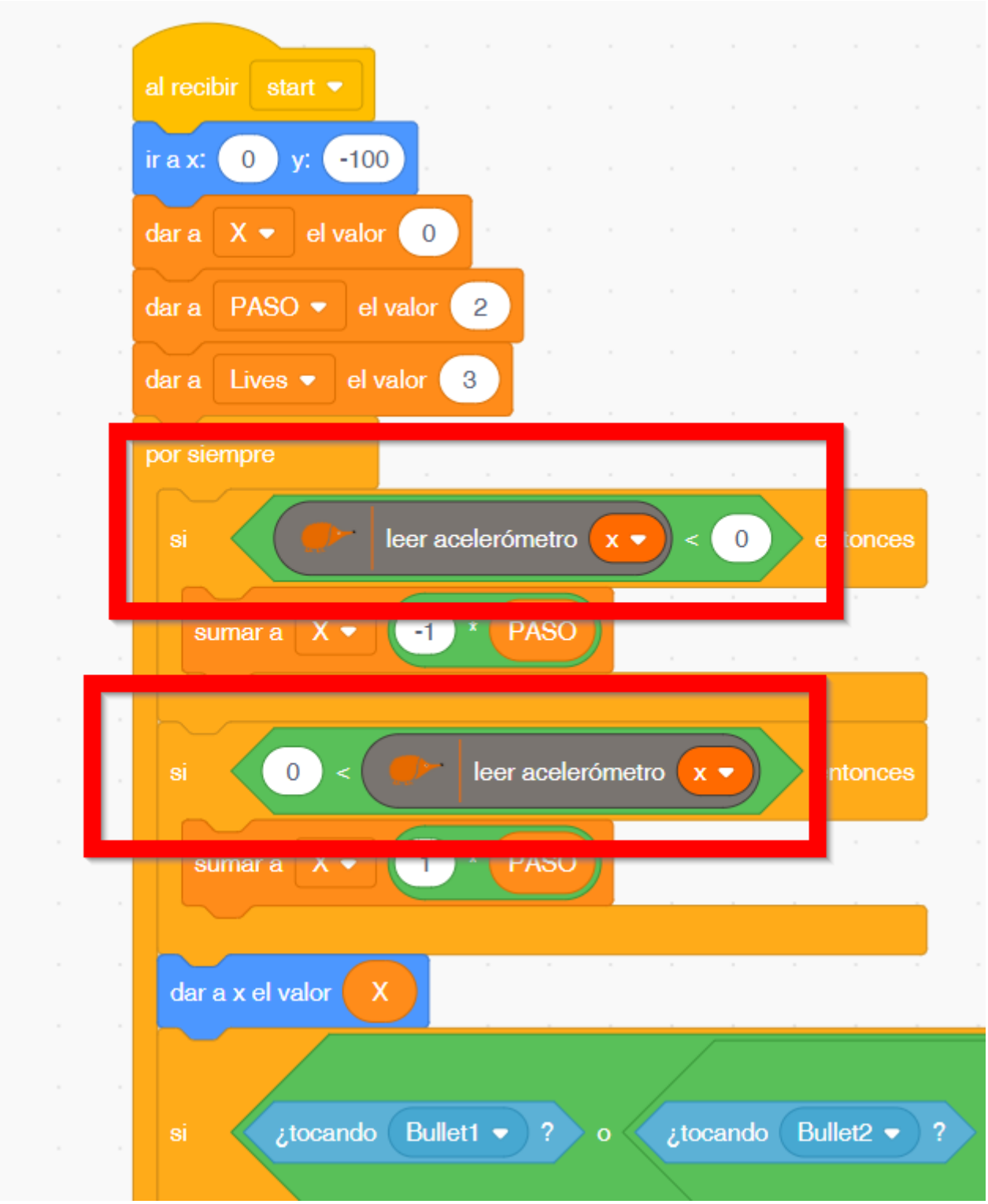
	VALOR MÍNIMO	VALOR REPOSO	VALOR MÁXIMO
JOYSTICK	0	512	1.024
ACELERÓMETRO	-1	0	+1

más o menos, pues cada placa es diferente

Puedes probar tus valores con este sencillo programa



Luego SI EL VALOR DEL ACELEROMETROX ES MAYOR QUE CERO que se mueva a la DERECHA y SI ES MENOR QUE CERO que se mueva a la IZQUIERDA así de sencillo tenemos que cambiar las instrucciones :



El programa lo tienes aquí https://github.com/JavierQuintana/Echidna/blob/master/MONTAJE%207-BIS%20ECHIDNABLACK2%20MATA-ALIENS%20CON%20ACELEROMETRO%20SIN%20MAPEAR.sb3

https://www.youtube.com/embed/DBJ8Str4sAA

Copyright 2026