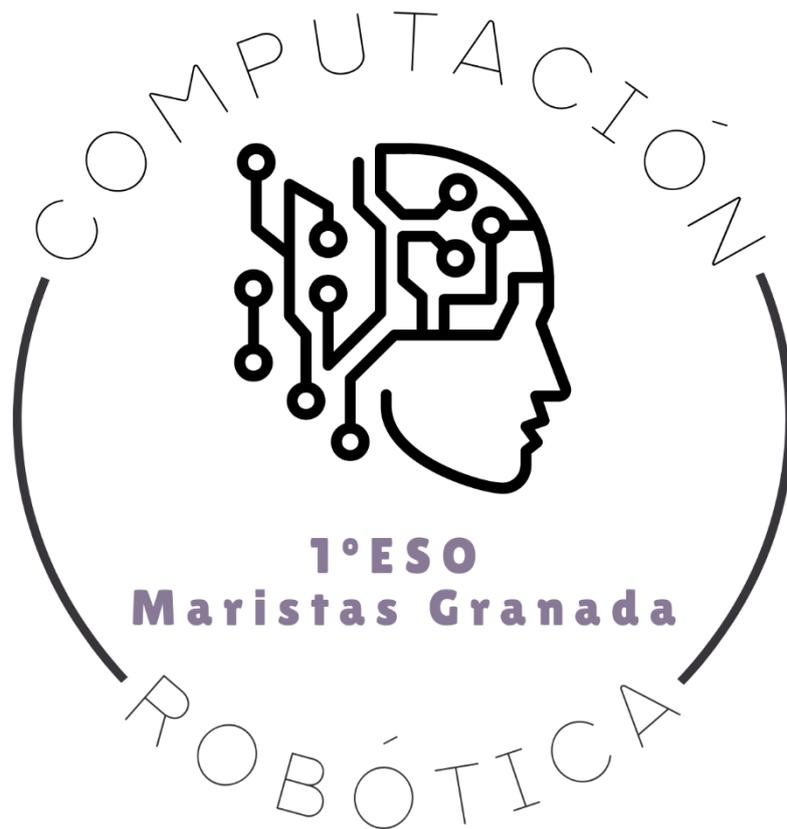


CURSO 2023-2024



# RETO 15: PRIMER VIDEOJUEGO CON SCRATCH

COMPUTACIÓN Y ROBÓTICA 1ºESO

COLEGIO MARISTA LA INMACULADA  
CALLE SÓCRATES, 8  
18002 - GRANADA

## Índice

0. Ubicación en la programación.....	2
1. Descripción del producto final a entregar.....	3
2. Aprendemos practicando .....	4
2.1 Videojuego paso a paso .....	4
3. Por si quieres seguir ampliando en casa .....	17

## 0. Ubicación en la programación

**Título:** Reto 15. Primer videojuego con Scratch.

**Evaluación:** Tercera

**Temporalidad:** 1 sesión (2 horas consecutivas)

**Índice de contenidos:** ..

**Breve resumen de la situación:** ..

## 1. Descripción del producto final a entregar

En este tema

**Los retos se califican según los siguientes criterios:**

- Están completos

## 2. Aprendemos practicando

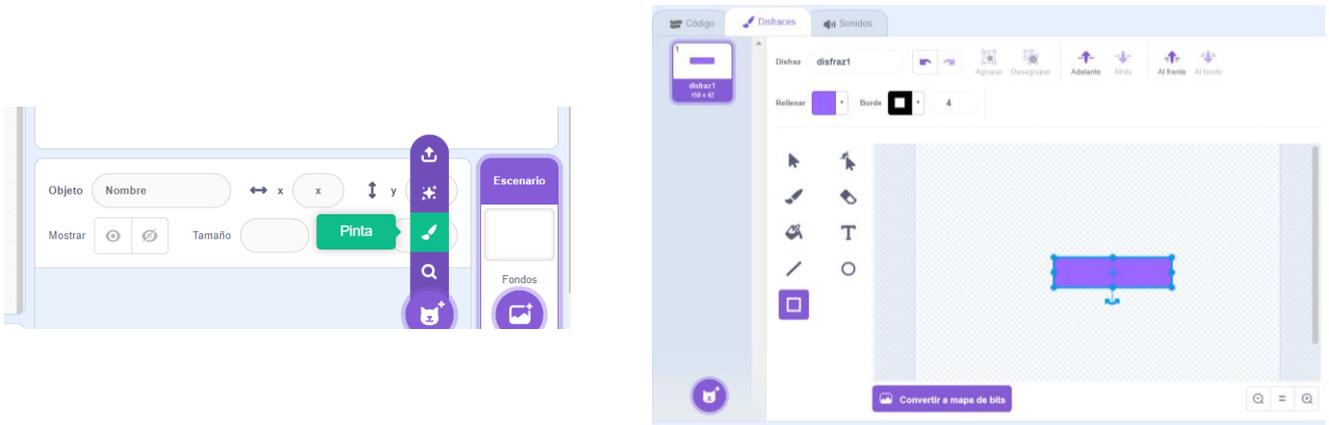
La siguiente actividad está inspirada en el siguiente vídeo, del canal de YouTube “Garaje Imagina”:

[https://www.youtube.com/watch?v=81fwfjIZgpE&ab\\_channel=GarajeImagina](https://www.youtube.com/watch?v=81fwfjIZgpE&ab_channel=GarajeImagina)

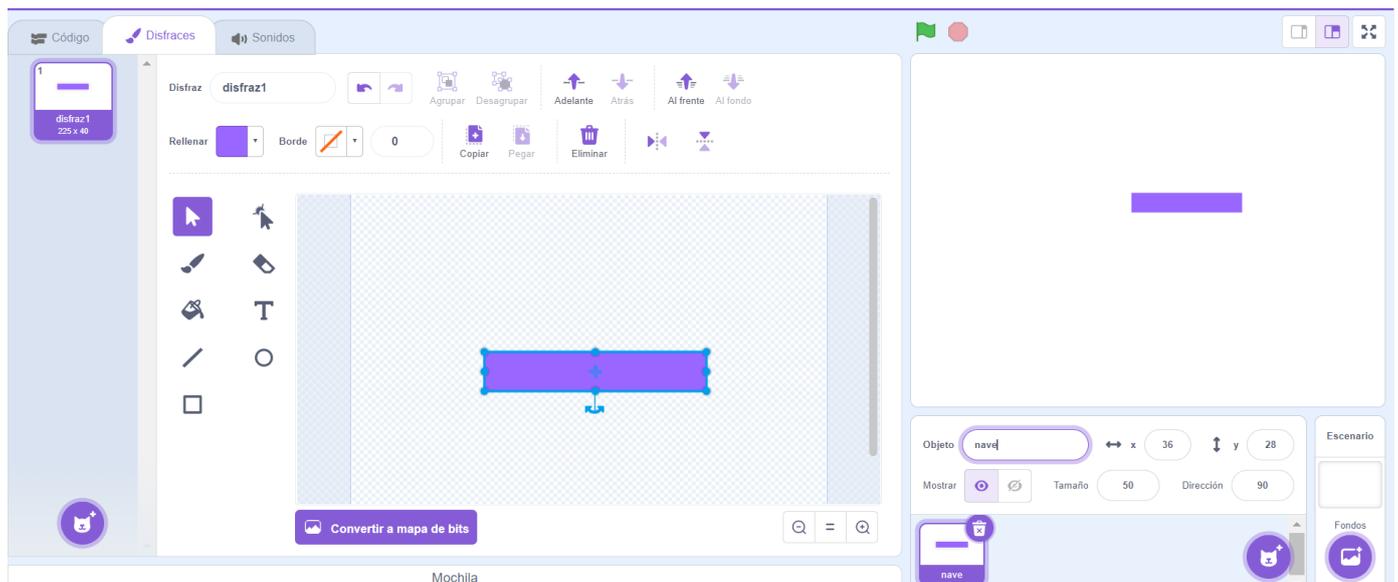
### 2.1 Videojuego paso a paso

Eliminamos el objeto por defecto (gatito) y creamos nuestra propia nave.

Por ahora, nuestra nave será un nuevo objeto (opción Pinta) que tendrá forma de rectángulo.

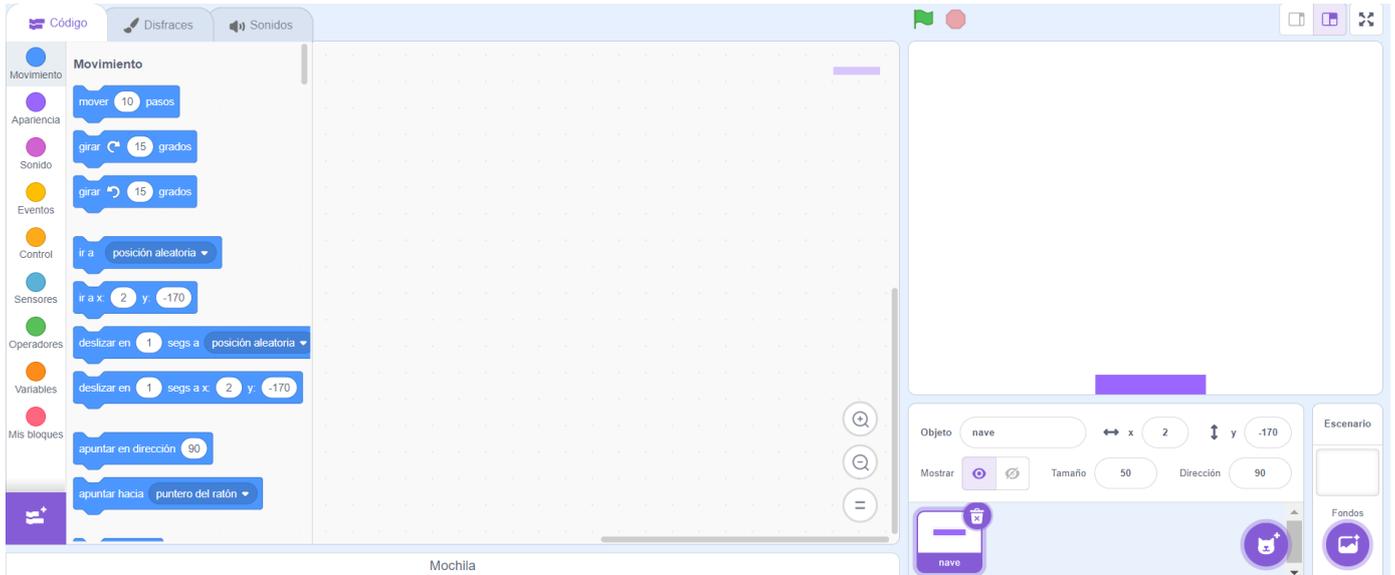


Eliminamos el borde del rectángulo y lo situamos justo en el centro. Será útil para el juego que el rectángulo sea bastante más ancho que alto. Nombramos al objeto como “nave”. La siguiente te sirve de referencia para ver el tamaño aproximado que tendrá este objeto en pantalla.



Pulsamos en la zona de código del objeto “nave” y colocamos el rectángulo en la parte inferior de la pantalla.

La nave se moverá de izquierda a derecha, permaneciendo siempre en la parte inferior de la pantalla.



Si pulso la flecha derecha, la nave se debe desplazar hacia la derecha.

Si pulso la flecha izquierda, la nave se debe desplazar hacia la izquierda.

Este movimiento lo puedo conseguir con el evento “Al presionar tecla” y con el bloque de Movimiento “sumar a x”.

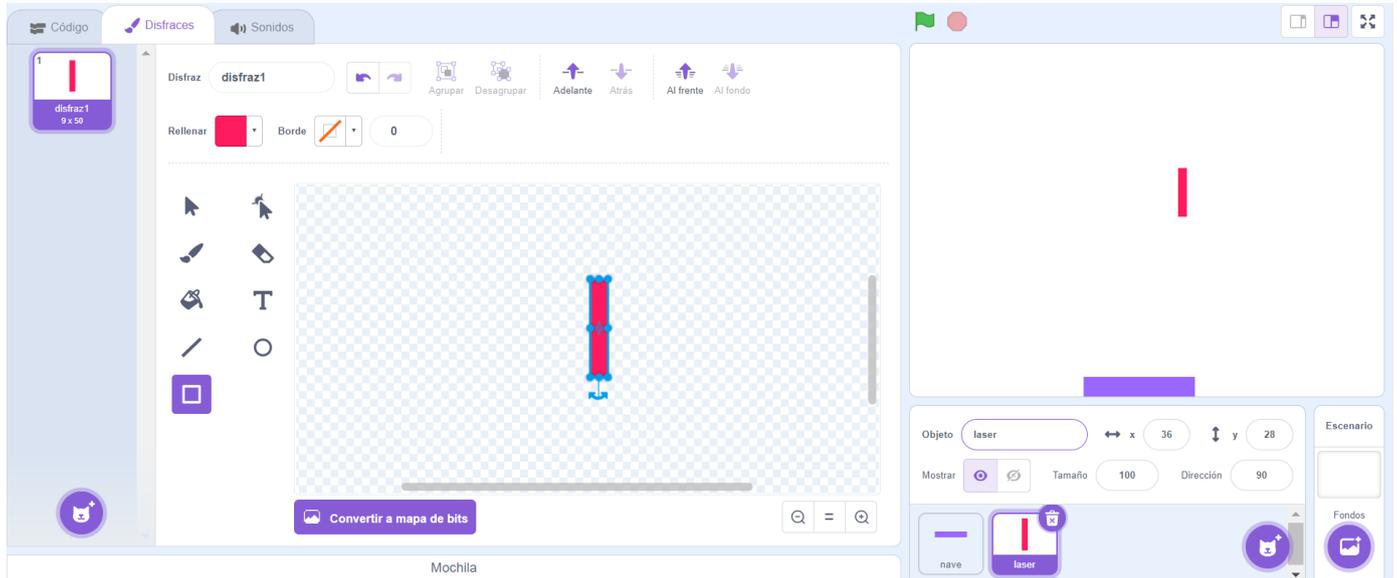


Nuestra nave disparará rayos láseres para destruir a los marcianitos.

Para ello, creamos un nuevo objeto que será un rectángulo vertical (más alto que ancho), también situado en el centro de la zona de dibujo y también sin borde.

A este nuevo objeto lo llamamos “laser”.

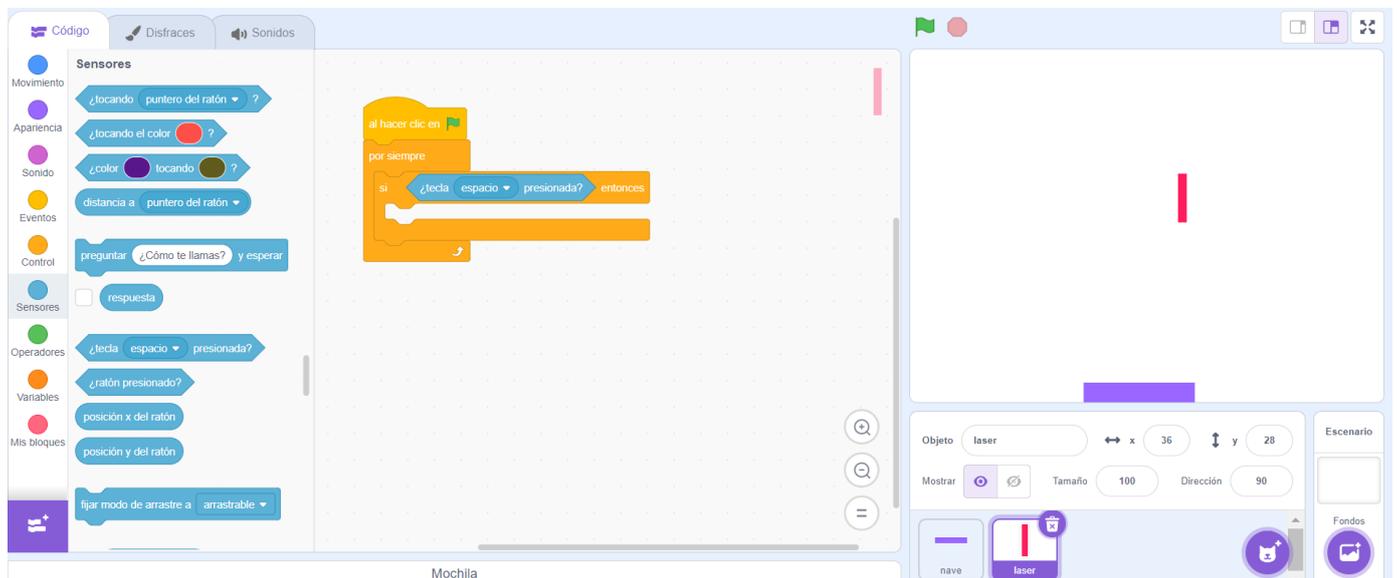
La siguiente imagen es una referencia aproximada del tamaño del láser. Es muy importante que, en la zona de dibujo, el rectángulo quede situado justo en el centro.



Como de costumbre, nuestro programa comenzará a funcionar cuando pulsemos la bandera verde.

Si pulsamos la bandera verde, queremos que al pulsar la tecla espacio nuestra nave lance un rayo láser. Para ello, necesitamos un bucle para siempre seguido de un condicional Si.

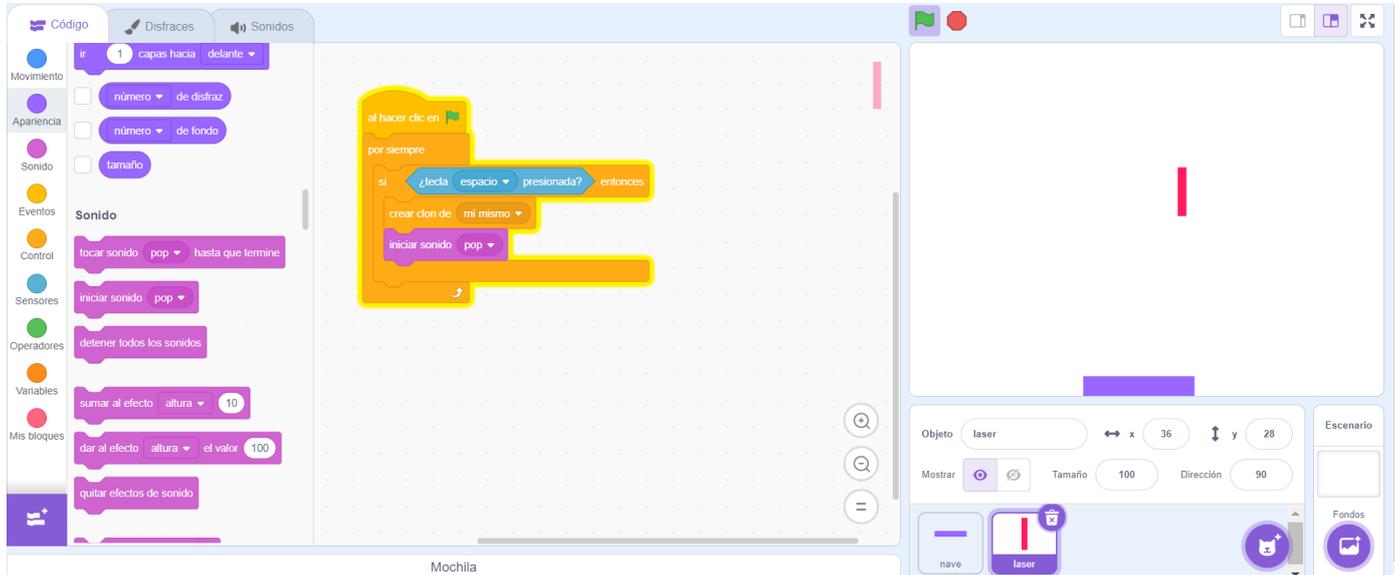
El siguiente código va asociado al objeto "laser".



Cada vez que pulse espacio, deseo lanzar un rayo. Tener que crear, uno a uno, todos los rayos que voy a lanzar puede hacerse infinito.

Para ahorrar tiempo y salud mental, vamos a utilizar el bloque "crear clon de" situado en las opciones de Control.

Cada vez que pulse espacio, crearé un clon del rayo láser y generaré un sonido de disparo.



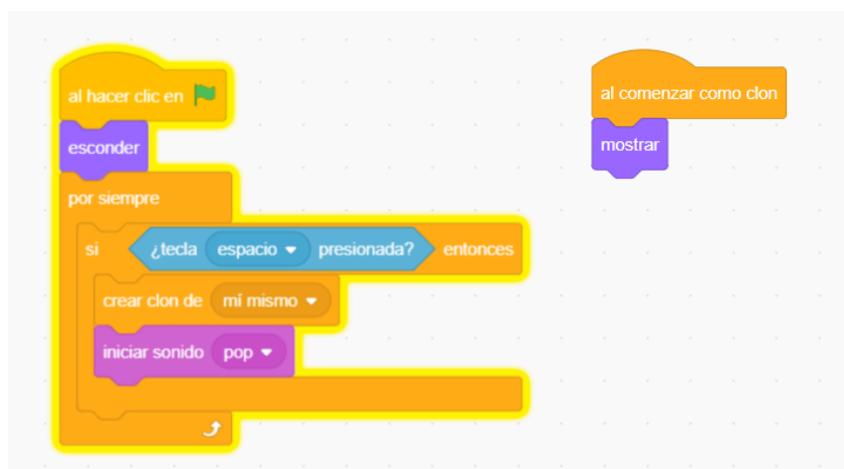
Ahora mismo, si pulsamos la tecla espacio, todos los clones se van creando superpuestos en la misma posición. Por eso no tenemos sensación de estar creando objetos nuevos.

Debemos programar dónde situar cada nuevo clon cada vez que pulsamos la tecla espacio.

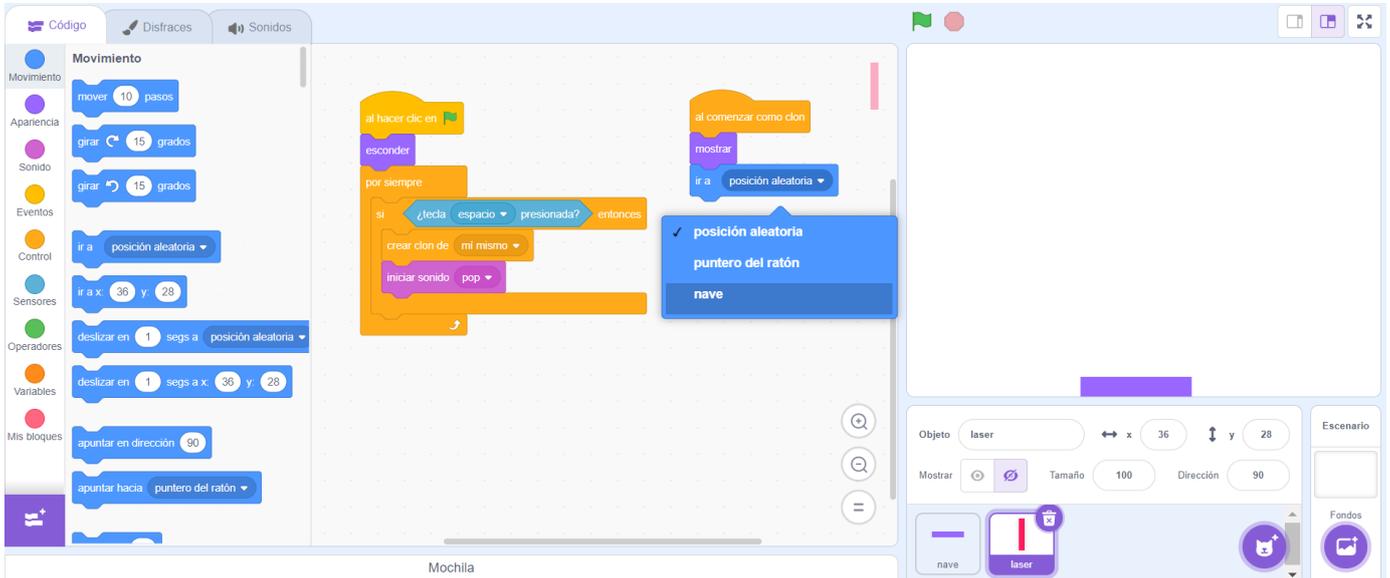
Cuando el juego comienza (tras pulsar bandera verde) el rayo debe estar oculto.

Solo mostraremos el rayo una vez que haya pulsado la tecla espacio. Por lo tanto, debo ocultar el objeto nada más iniciarse el juego. Y debo mostrarlo a continuación de que se ejecute el bloque “al comenzar como clon”.

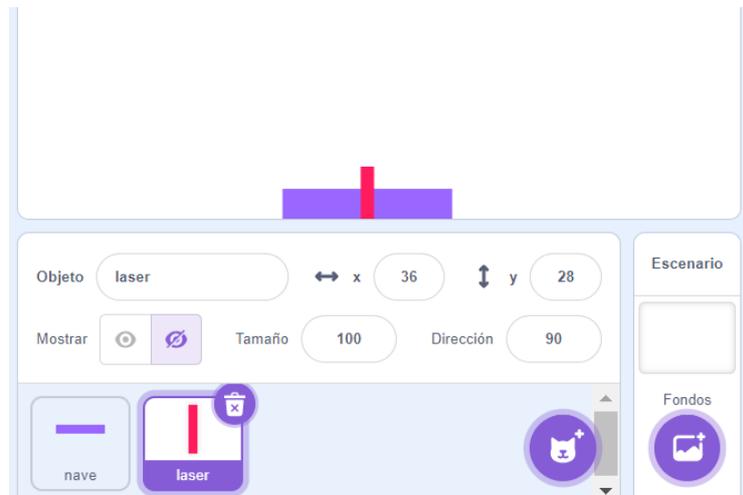
Fíjate en la siguiente imagen.



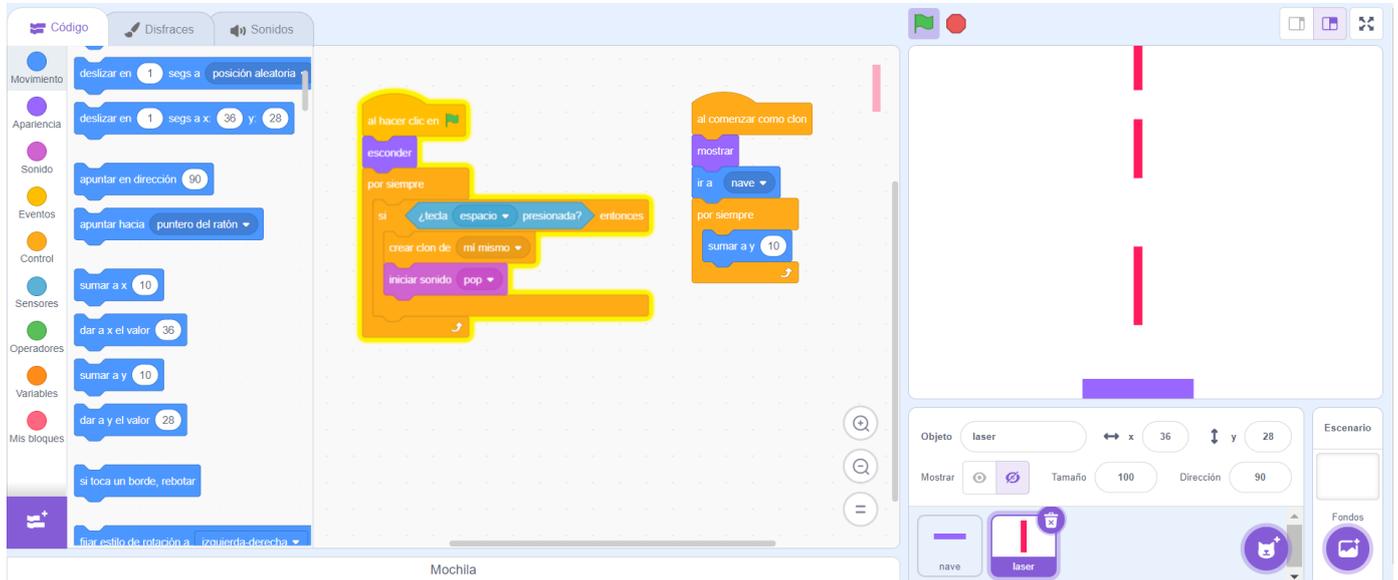
Cuando pulse la tecla espacio, el rayo láser debe salir de la nave. Por lo tanto, cuando se ejecuta “al comenzar como clon” debo llevar al clon a la posición de la nave. Dentro de la opción Movimiento encontramos un bloque “ir a” que nos permite situar un objeto en la posición central de otro objeto. Nosotros elegiremos “ir a nave”.



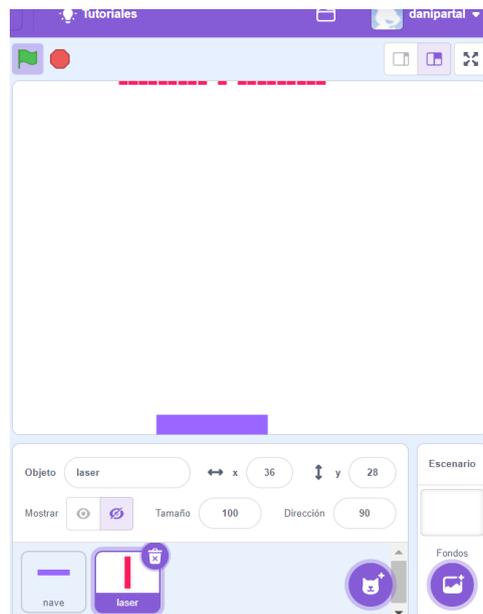
Con esto, cada vez que pulsemos la barra espacio, los objetos “laser” se situarán en el centro de nuestra nave.



No quiero que el rayo se quede siempre en la nave. Necesito que el rayo suba verticalmente. Por lo tanto, usamos un nuevo bucle para siempre que desplaza la coordenada vertical (coordenada y) del objeto “laser”.



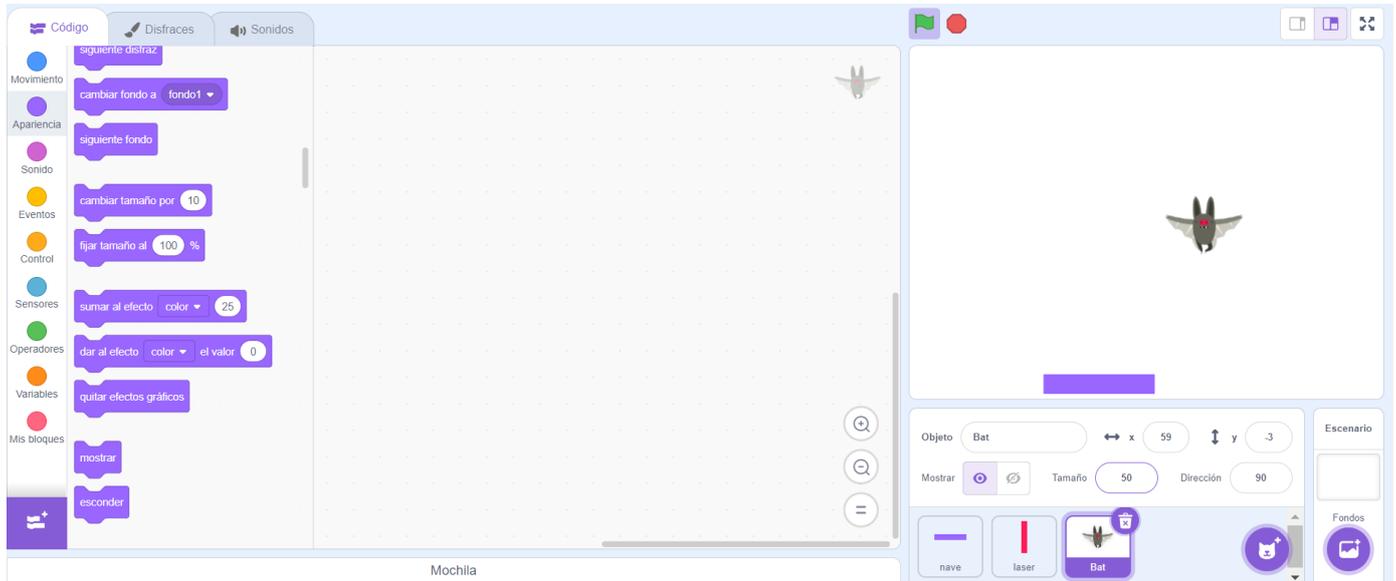
¡Problema! Los rayos se quedan almacenados en la parte superior. Y eso queda bastante feo.



¿Cómo resolverlo? Con un condicional Si que controle que, si el rayo toca el borde, desaparezca. ¡Ojo! Separa un poco la nave de la base inferior de la pantalla, para que el rayo no toque el borde de la parte inferior y desaparezca desde el inicio. El rayo solo debe desaparecer al tocar el borde superior de la pantalla.

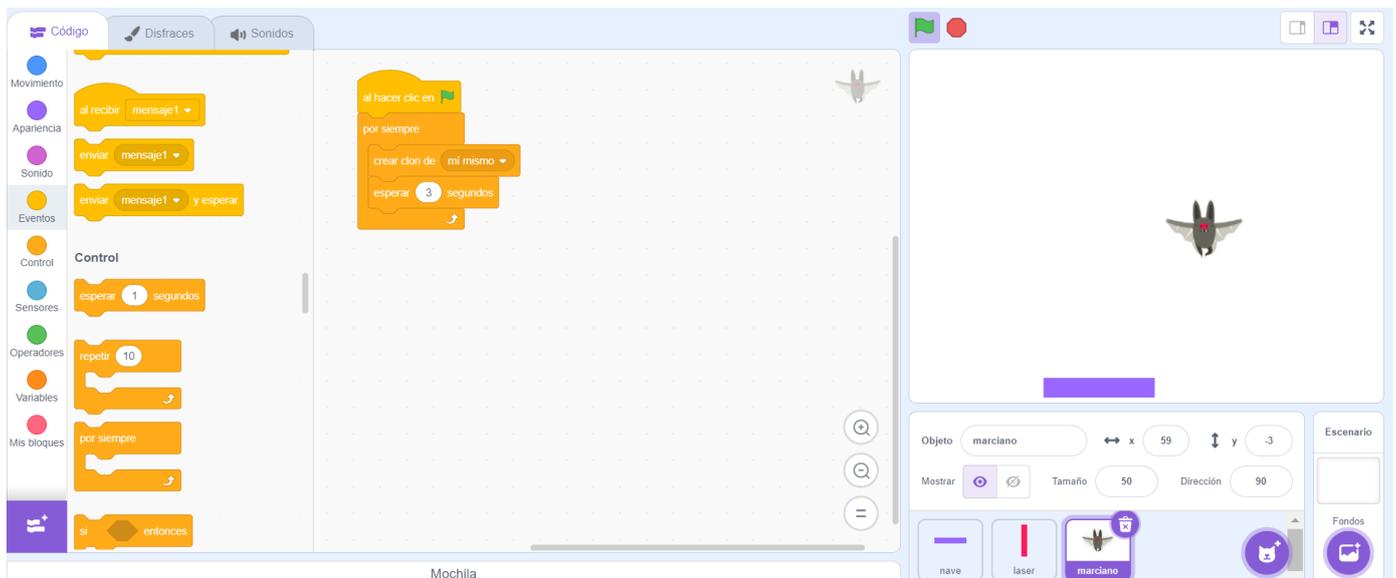


Como marcianito (o enemigo) vamos a usar un objeto de las ya disponibles en Scratch. Usaremos el murciélago. Posee varios disfraces por defecto, y es sencillo dar la apariencia de estar volando.



Al objeto murciélago le damos por nombre “marciano”.

Lo primero que vamos a programar para “marciano” es lo siguiente: desde el inicio del juego (tras pulsar bandera verde) se crearán clones cada 3 segundos.

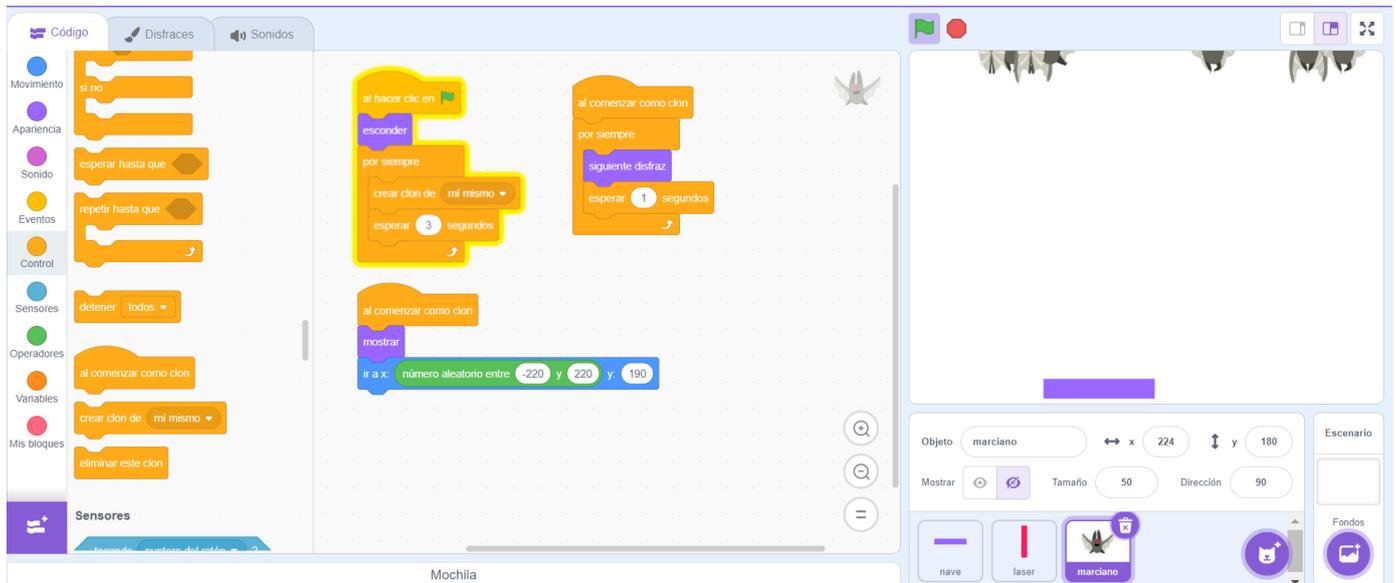


Además, para generar sensación de volar, vamos a cambiar cada 1 segundo el disfraz vinculado al objeto. Usamos el bloque “al comenzar como clon”.

Nuevamente, cuando el juego se inicia el marciano debe estar oculto. Y solo se mostrará al “comenzar como clon”.

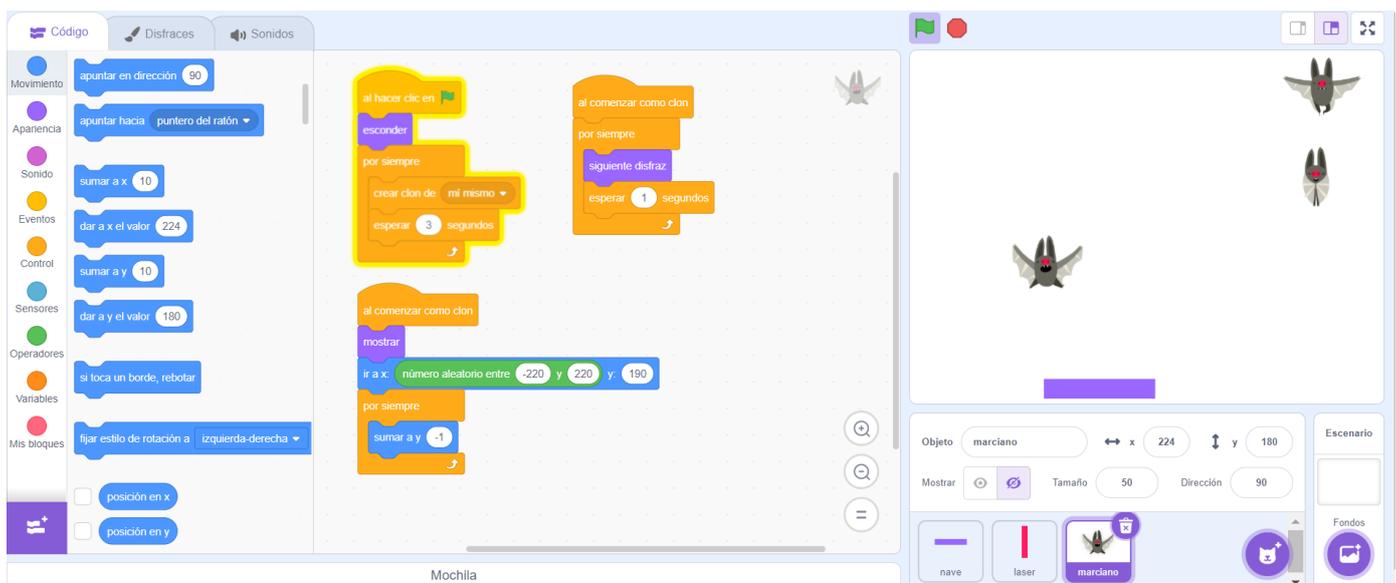
Además, deseamos que el marciano aparezca por la parte superior y vaya cayendo verticalmente. Para situarlo en una posición aleatoria de la parte superior de la pantalla, podemos fijar la coordenada vertical (coordenada y) al valor 190.

Y podemos decidir que la coordenada horizontal (coordenada x) sea un número aleatorio entre -220 y 220. Estos valores salen de situar el murciélago por los bordes de la pantalla y mirar los valores de sus coordenadas.



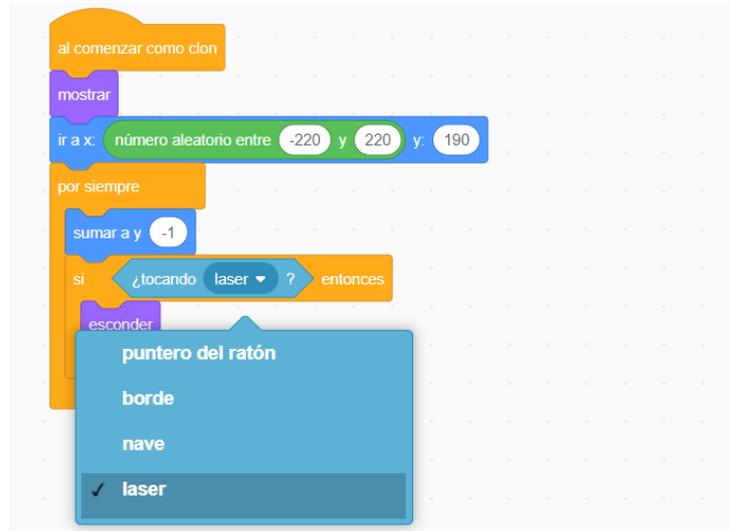
En la imagen anterior, los marcianos aparecen volando en distintos puntos del borde superior pero no caen verticalmente. Debemos crear ese movimiento vertical.

Usamos un nuevo bloque “para siempre” y un bloque “sumar a y”. Para que el marcianito baje lento, podemos escribir la cantidad negativa -1 en “sumar a y”.



¿Qué debe ocurrir si el láser impacta con un marcianito? Pues que el marcianito debe desaparecer.

¿Cómo lo conseguimos? Con un condicional Si que controle si cada clon de “marciano” toca a un objeto “láser”.



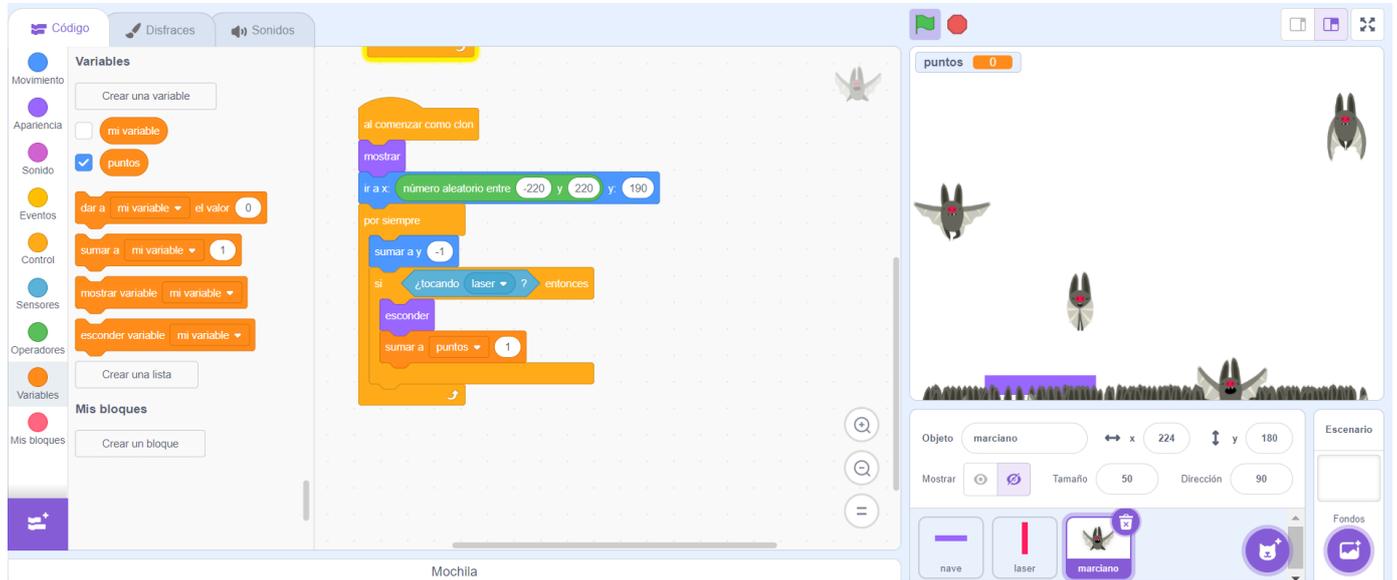
Con una variable vamos a contabilizar el número de marcianitos que eliminamos. Llamamos a esa variable “puntos”.



Cuando el juego comienza, tengo 0 puntos. Añado este dato en el código de programación de los marcianitos.



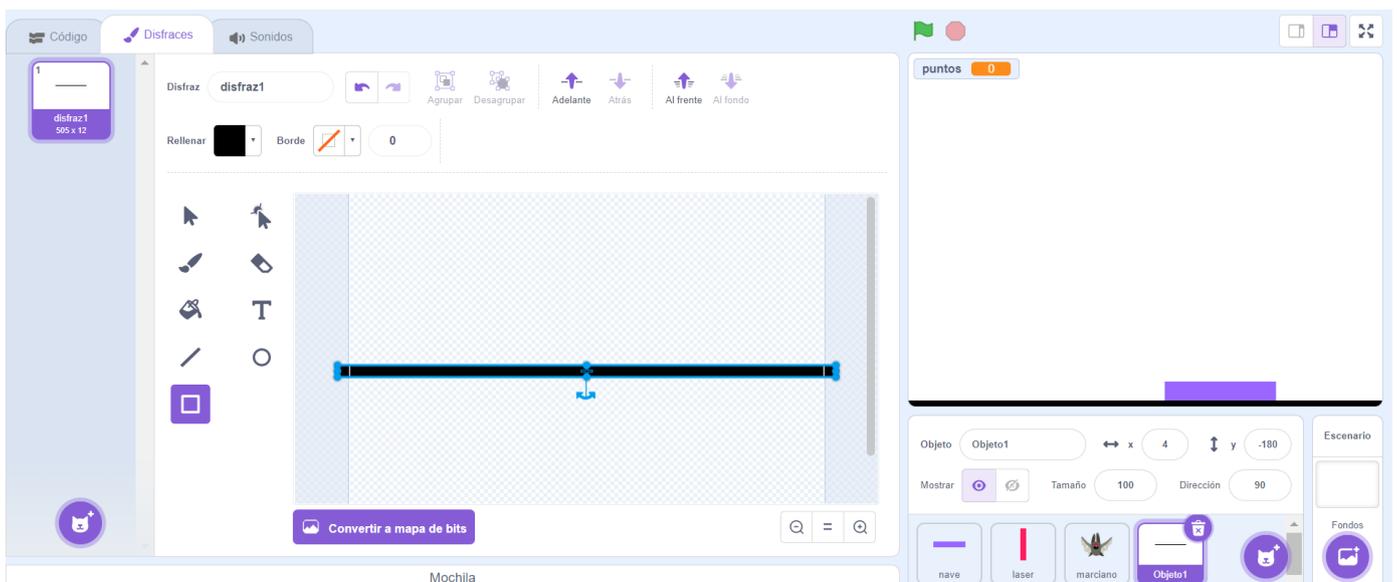
Cada vez que el láser impacte con un marcianito, debo sumar 1 a la variable “puntos”. Por lo tanto, justo en el código donde detectaba el impacto entre “láser” y “marciano” añado un bloque para sumar 1 al valor de la variable “puntos”.



En la imagen anterior, a la derecha, se aprecia como los marcianitos se están acumulado en la parte inferior del borde del juego.

Voy a hacer lo siguiente: si un marcianito llega abajo, debe desaparecer y además me debe penalizar restando vidas. Y si pierdo todas las vidas disponibles, el juego terminaría.

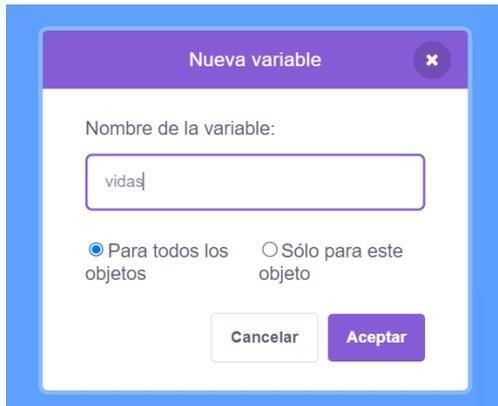
¿Cómo detectar si el marcianito llega al suelo? Puedo crear un suelo de color negro, por ejemplo, y detectar si un objeto "marciano" impacta con el objeto "suelo".



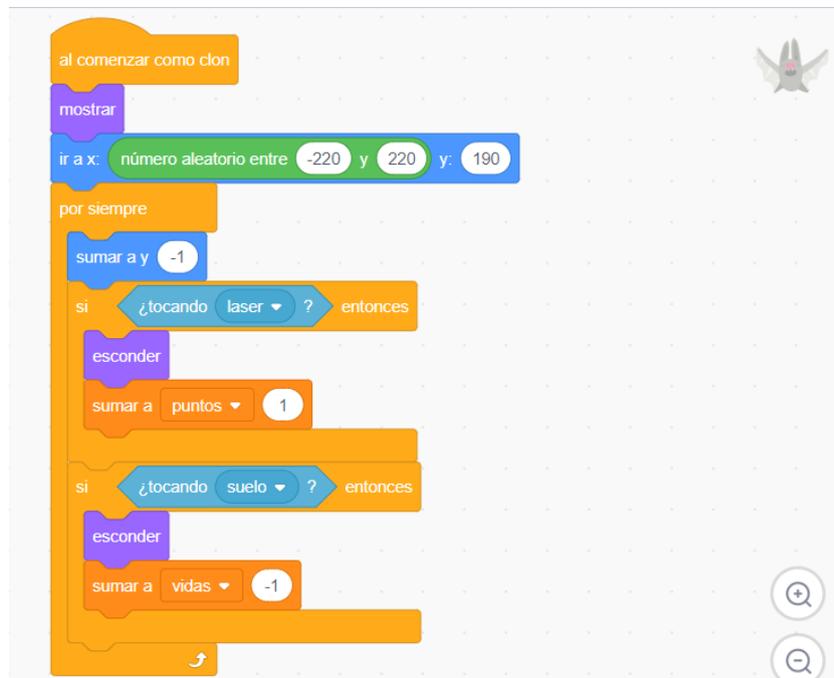
Creamos una variable "vidas" que inicialmente toma el valor 3.

Cuando un marcianito llegue al suelo, restaremos una vida.

d

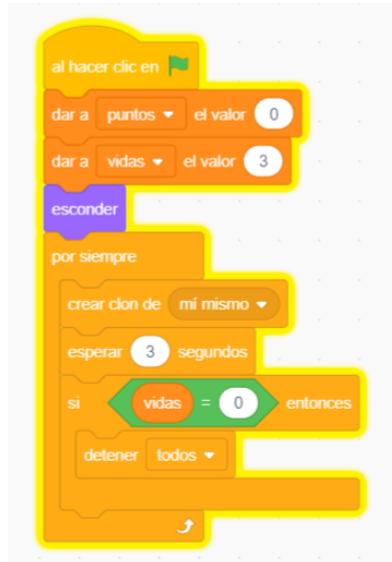


Cuando el marcianito impacta con el suelo, debe desaparecer y debe restar una unidad a la cantidad de vidas.

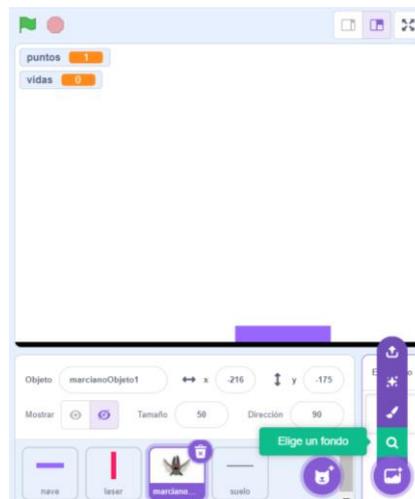


¿Qué ocurre cuando el número de vidas es igual a 0? El juego se termina.

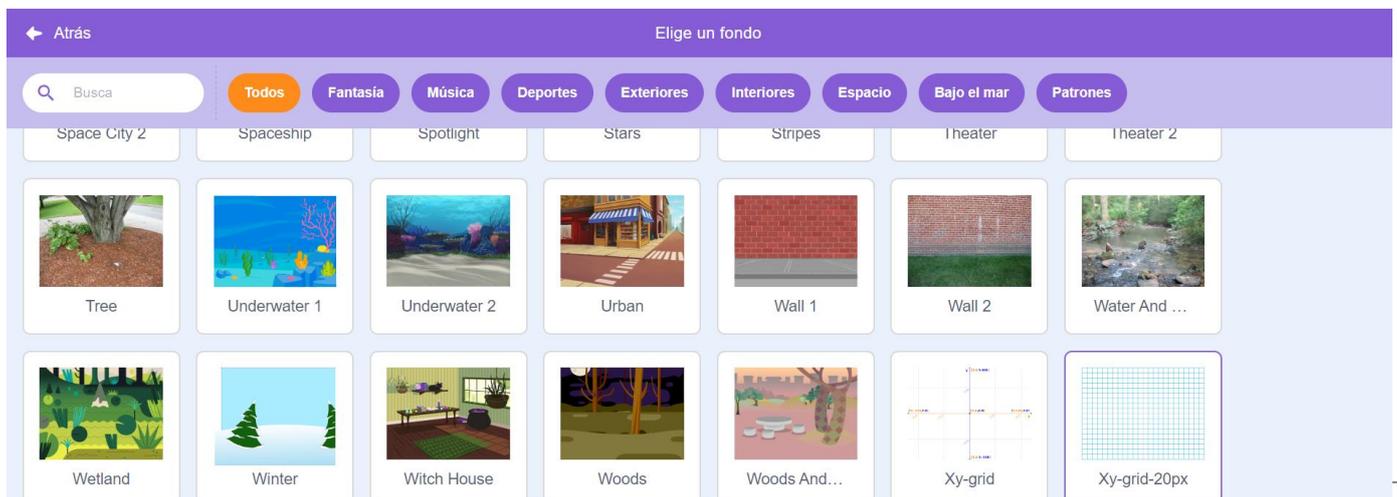
Un condicional Si nos advierte fácilmente del fin del juego. Fíjate en cómo modificamos el código asociado al objeto "marciano". Si "vidas" es igual a 0, debemos detener todos los objetos del programa.



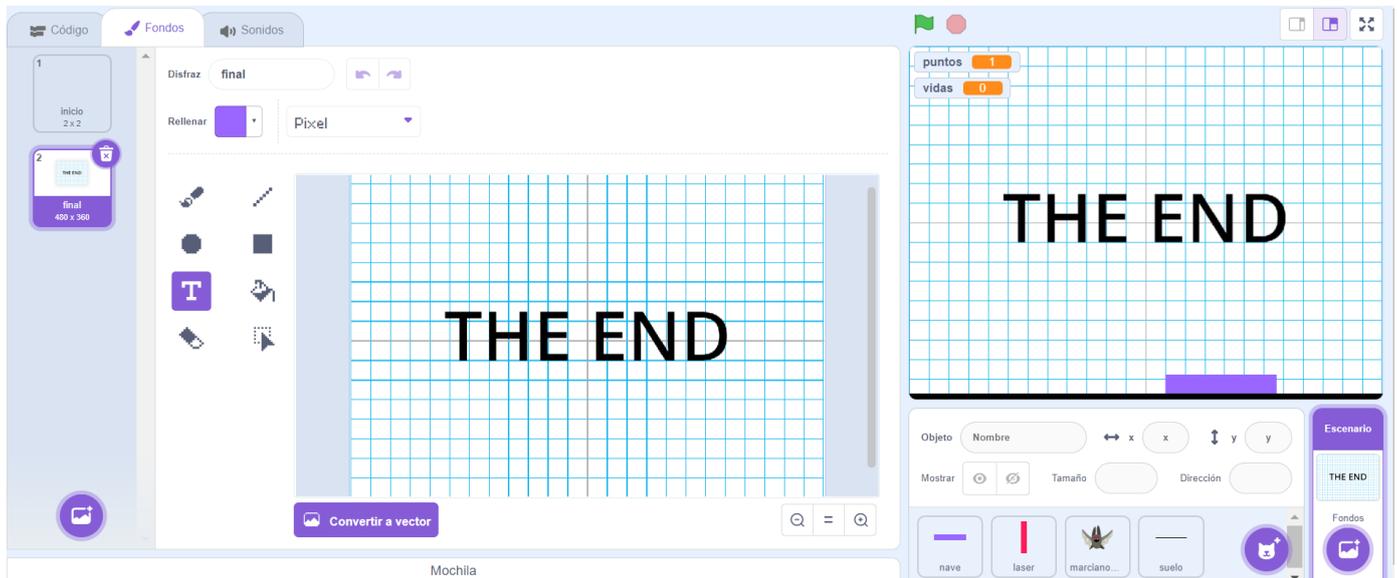
Cuando el juego termine, vamos a cambiar el fondo. Informaremos del fin (“The End”).



Elegimos, por ejemplo, el fondo de cuadrículas que aparece por defecto en las opciones de elegir fondo.



Al primer fondo lo podemos llamar “inicio” y al segundo fondo lo llamamos “final”. En “final” podemos añadir las palabras “The End”.



En el código de programación del objeto “marciano”, al pulsar bandera verde, podemos mostrar el fondo “inicio” al comenzar el juego. Y cuando se acabe la partida, y todos los objetos se detengan, podemos mostrar el fondo “final”.



Ya lo tendríamos. Ahora es el momento de jugar. Recuerda que puedes variar la velocidad de caída modificando el valor numérico del bloque “sumar a y” del objeto “marciano”.

### 3. Por si quieres seguir ampliando en casa

....