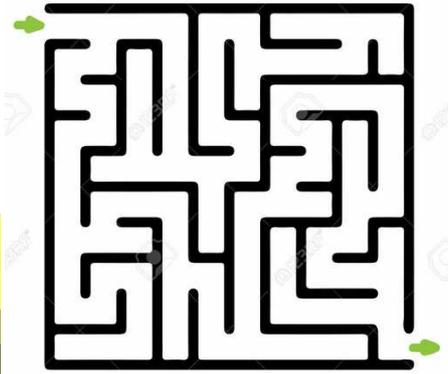
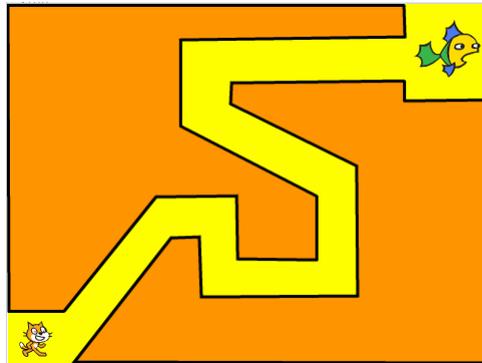


## ACTIVIDADES PARA SCRATCH

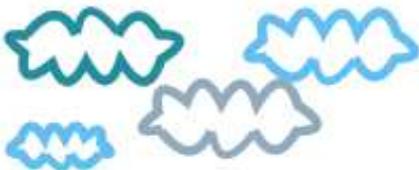
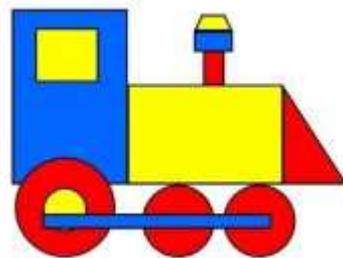
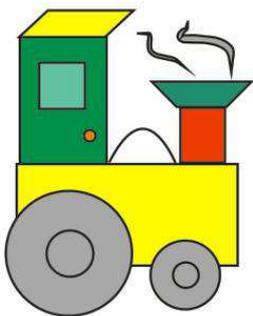
### 1.- El editor de dibujos

1.1.- Dibuja con el editor de dibujo de Scratch un fondo para el escenario con el siguiente laberinto. Hazlo en modo mapa de bits. Lo usaremos para una actividad posterior.

1.2.- Dibuja con el editor de Scratch el siguiente fondo para una actividad posterior. Usa el modo vector

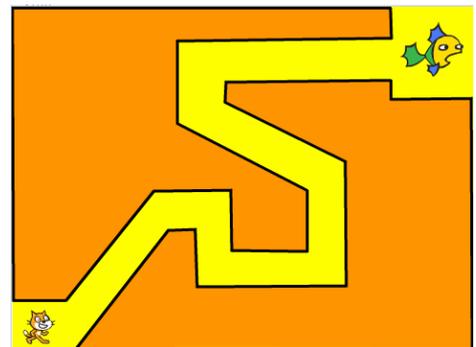


1.3.- Dibuja con el editor de imágenes los siguientes objetos.



### 2.- El movimiento de los objetos

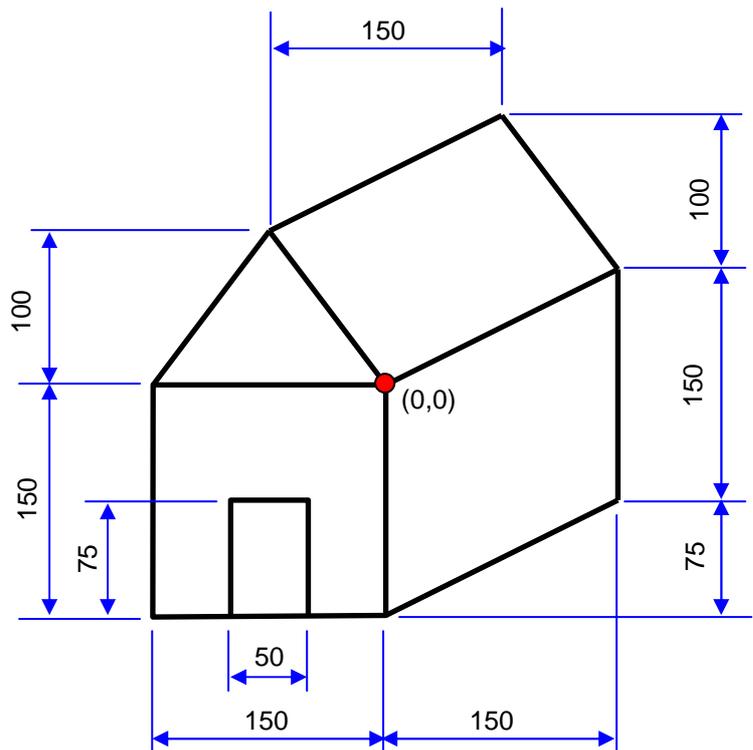
2.1.- Realiza un programa para que, al hacer clic en la bandera verde el gato haga el recorrido por el camino amarillo dibujado en la actividad 1.2 hasta llegar al pez (que es un objeto de la biblioteca de Scratch). El gato debe ir dibujando su recorrido. La línea dibujada por el gato no debe salir nunca del camino dibujado en amarillo. Al presionar la tecla "espacio" debe borrarse la línea dibujada y el gato debe de nuevo al inicio.



**2.2.-** Realiza un programa para que, al hacer clic en la bandera, el gato dibuje la figura con forma de casa que se indica. El gato debe empezar y terminar en el punto donde se ha colocado el círculo rojo, que tendrá las coordenadas (x=0, y=0) de la pantalla de Scratch.

Para hacer el dibujo, primero deberás calcular las coordenadas de cada uno de los vértices de la figura a partir de las cotas que se indican. Una vez las tengas, el programa debe hacer que el gato vaya de unas coordenadas a otras para ir dibujando las líneas.

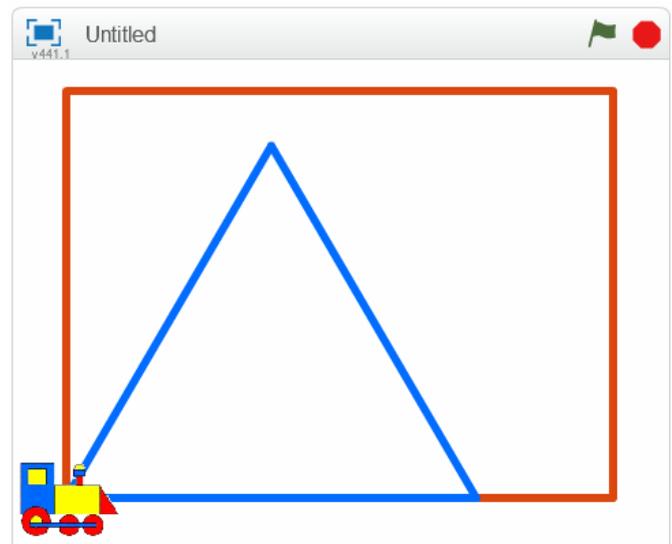
Al pulsar la tecla “b” se borrará todo.



### 3.- El lápiz

**3.1.-** Realiza un proyecto para que, al hacer clic sobre el objeto (una de las locomotoras del ejercicio anterior o, si no, el gatito de Scratch), empiece situándose ésta en la esquina inferior izquierda, a continuación, realice un rectángulo de 400 pasos de ancho y 300 de alto de un color determinado, después cambiará de color y hará un triángulo de 300 pasos de lado.

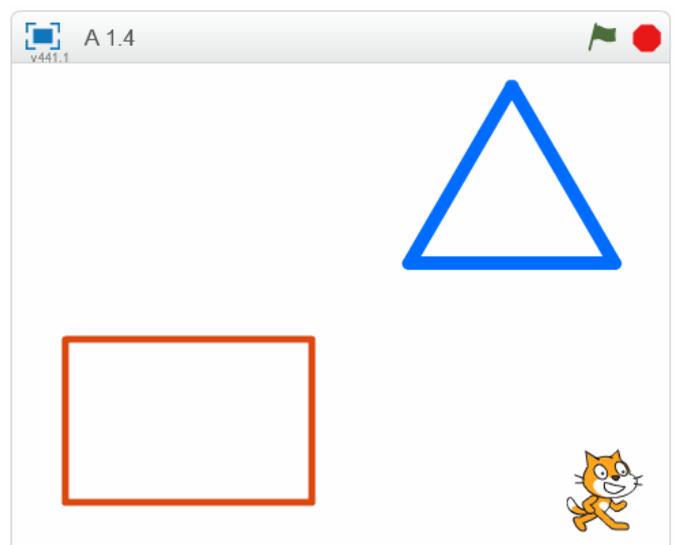
Cuando presionemos la tecla espacio, se borrarán el rectángulo y el triángulo y la locomotora se quedará en la esquina inferior a la espera de que volvamos a hacer clic sobre ella. El proyecto tendrá dos programas.



**3.2.-** Realiza un proyecto para que el gato realice los siguientes dibujos en la pantalla. Al hacer clic sobre el gato, éste ira a la esquina inferior izquierda, donde debe iniciar el programa, a continuación debe realizar los dibujos y, por último, desplazarse a la esquina inferior derecha, donde terminará la animación.

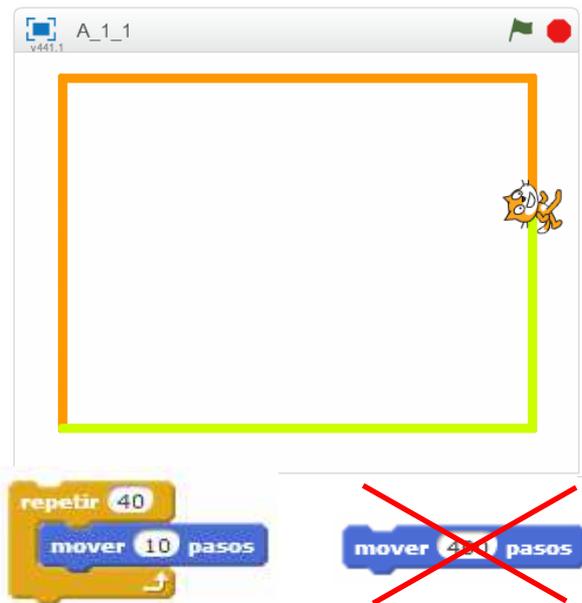
Observa que tanto el color como el grosor del lápiz son diferentes en el triángulo y en el rectángulo.

Al teclear la letra “b” borrará todo y volverá a su esquina inferior izquierda.



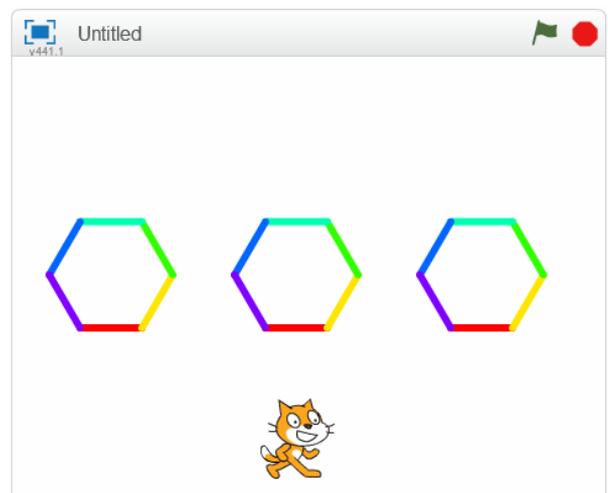
#### 4.- Los bucles repetitivos

**4.1.-** Realiza un proyecto para que el gato se sitúe en la esquina inferior izquierda de la pantalla y empiece a moverse realizando rectángulos de 400 pasos de ancho por 300 de alto indefinidamente hasta que se pulse la tecla “p”. Al terminar cada rectángulo el gato dirá “miau” durante un segundo en una burbuja de texto (no sonido pues en el aula no hay altavoces). Cada vez que inicia un nuevo rectángulo cambiará el color del lápiz. Consigue que el programa no se ejecute a tal velocidad que impida visualizar el movimiento del gato (bucles Repetir).



**4.2.-** Realiza un programa para que el gato realice el dibujo de la figura y después se vaya a la parte inferior de la pantalla, como queda en la figura.

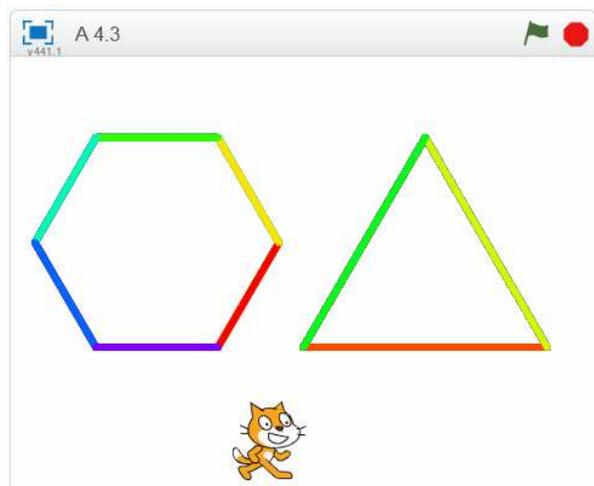
Utiliza como grosor del lápiz 6 puntos y ve cambiando el color en 30 unidades al pasar de un lado a otro de los hexágonos.



**4.3.-** Realiza un programa para que el gato realice el dibujo de la siguiente forma:

Primero el gato girará dibujando el hexágono una y otra vez cambiando de color constantemente hasta que presionemos la letra “a”. Entonces empezará a girar dibujando una y otra vez el triángulo, cambiando de color constantemente hasta que presionemos la letra “b”. Entonces parará y se situará en parte inferior de la pantalla como se ve.

El hexágono tiene 100 pasos de lado y el triángulo 200 pasos. El grosor del lápiz es de 6.



## 5.- La apariencia de los objetos

**5.1.-** Realiza un programa que haga lo siguiente: al hacer clic en la bandera verde el gato, situado inicialmente en el lateral izquierdo de la pantalla, se desplazará hacia la derecha 200 cambiando de disfraz para parecer que camina. A continuación se detendrá, dirá "Voy a descansar" y esperará parado indefinidamente hasta que lo toquemos con el puntero del ratón (sin hacer clic). Entonces dirá "Empezaré de nuevo a caminar" por 2 segundos y volverá a desplazarse cambiando de disfraz otros 200 pasos.

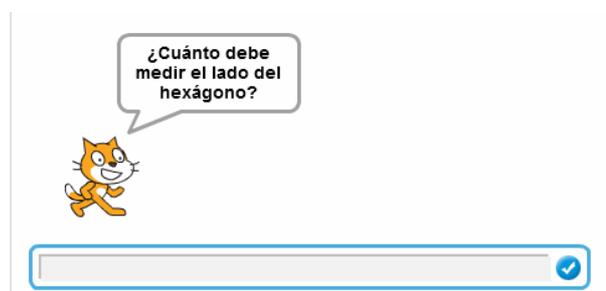


**5.2.-** Abre un archivo de Scratch. Carga el fondo "space" de la biblioteca. Haz una copia y modifica la imagen para que, en vez de noche sea de día. Carga los botones desde la biblioteca de Scratch y añádeles los letreros Día y Noche. Carga la bailarina de la biblioteca de Scratch. El programa debe funcionar haciendo que, al hacer clic en la bandera verde, la bailarina empiece a bailar pero a un ritmo no demasiado rápido y el fondo del escenario debe cambiar de día a noche cada vez que hagamos clic sobre el botón correspondiente.



## 6.- La entrada de datos por teclado

**6.1.-** Haz un programa para que, al hacer clic en la bandera verde, el gato dibuje un hexágono, como en otras ocasiones, pero ahora, antes de hacerlo, preguntará cuánto debe medir el lado del hexágono.

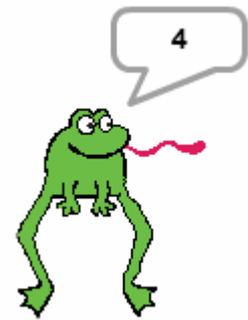


**6.2.-** Haz un programa para que, al hacer clic en la bandera verde, la rana pregunte cuántos saltitos hacia arriba queremos que dé. Tras darle este número, la rana dará dicho número de saltos a intervalos de 1 segundo.



## 7.- Las variables y las listas

**7.1.-** Haz un programa para que, a partir de que hagamos clic en la bandera verde, la rana dé un salto cada vez que presionemos la tecla “espacio” pero que cuando esté arriba diga el número de saltos que lleva dados (de momento sólo tiene que decir el número). Trata de hacerle un nuevo disfraz a la ranita para que parezca que ha dado el salto (más o menos como en la figura).



**7.2.-** En la actividad 4.1 el gato se mueve realizando rectángulos de 400x300 pasos hasta que se pulsa la tecla “p”. Modifica el programa para que, al empezar, nos pregunte el ancho y el alto de los rectángulos y para que en el momento en que lo paremos, pulsando la tecla “p”, el gato nos diga cuántos rectángulos lleva hechos.



**7.3.-** Realiza una nueva modificación del programa de la actividad 4.1 de modo que cuando pulsemos la tecla “p” estando el gato realizando un rectángulo, no se pare de inmediato sino que espere a que termine el rectángulo que está haciendo en ese momento y, una vez lo acabe, entonces se pare y diga cuántos ha hecho. No debe ser necesario mantener pulsada la tecla “p”, basta una breve pulsación (te aconsejo una solución: definir una variable que cambie de valor al pulsar la tecla “p”, y que se consulte el valor de dicha variable en el momento de empezar cada nuevo rectángulo).

**7.4.-** Haz un programa para que al hacer clic en la bandera verde, el pingüino haga al usuario varias preguntas (el nombre, la edad, la ciudad donde vives y el deporte favorito). Tras realizar todas las preguntas dirá una frase con todos esos datos. Por ejemplo: “Hola Juan, tienes 15 años, vives en Sevilla y tu deporte favorito es Baloncesto”.

Ayuda: Te recomiendo que crees cuatro variables para almacenar cada uno de los cuatro datos que preguntas, o bien una lista con cuatro elementos para guardar dichos datos.



**7.5.-** Hacer un proyecto en el que crees tres listas, una con las provincias de Andalucía, otra con las de Galicia y otra con las de Aragón. En pantalla aparecerán tres personajes. Cada uno tendrá asignada una comunidad. Al hacer clic en cada uno de ellos, el personaje dirá primero la comunidad que le corresponde y a continuación todas las provincias de dicha comunidad, a intervalos de 1,5 segundos entre una y otra.



## 8.- Las operadores matemáticos, relacionales y lógicos.

8.1.- Haz un proyecto con los personajes de la figura de modo que funcione del siguiente modo: al hacer clic sobre uno de los personajes preguntará por dos números, o bien uno en el caso de la raíz cuadrada. A continuación te dirá el resultado de aplicar la operación correspondiente a dicho personaje.

Los mensajes tendrán forma de frase. Por ejemplo, "la multiplicación de 15 y 18 es 270".

Intenta conseguir que la raíz cuadrada y la división te la diga redondeada pero a un único decimal.



## 9.- La toma de decisiones.

9.1.- Vamos a cambiar un poco el programa anterior. Ahora el funcionamiento será el inverso: al hacer clic sobre uno de los personajes preguntará la operación matemática correspondiente (la que pone junto a ellos) con números al azar en los siguientes rangos:

- Las sumas y las restas con números entre 1 y 100.
- Las multiplicaciones con números entre 1 y 20.
- En las divisiones el dividendo se rá entre 1 y 500 y el divisor entre 1 y 10. El resultado se dará redondeado al número entero más próximo.
- Las raíces cuadradas serán de números entre 1 y 700 pero debes asegurarte de que pregunte siempre un número cuya raíz cuadrada sea un número exacto.



El programa llevará un contador de preguntas y un contador de aciertos, que se mostrarán en la parte superior de la pantalla, así como el porcentaje de aciertos (con redondeo a un número decimal). Estos contadores se pondrán a 0 cuando se haga clic sobre la bandera verde.

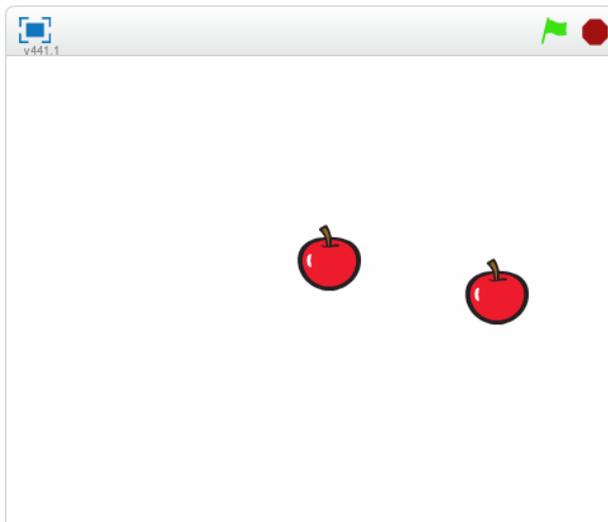
Intenta la siguiente mejora: El número máximo de preguntas será 10. Una vez superado este número ya no se seguirá preguntando hasta que se haga clic con la bandera verde y se pongan a 0 de nuevo los contadores.

## 10.- Las interacción entre objetos.

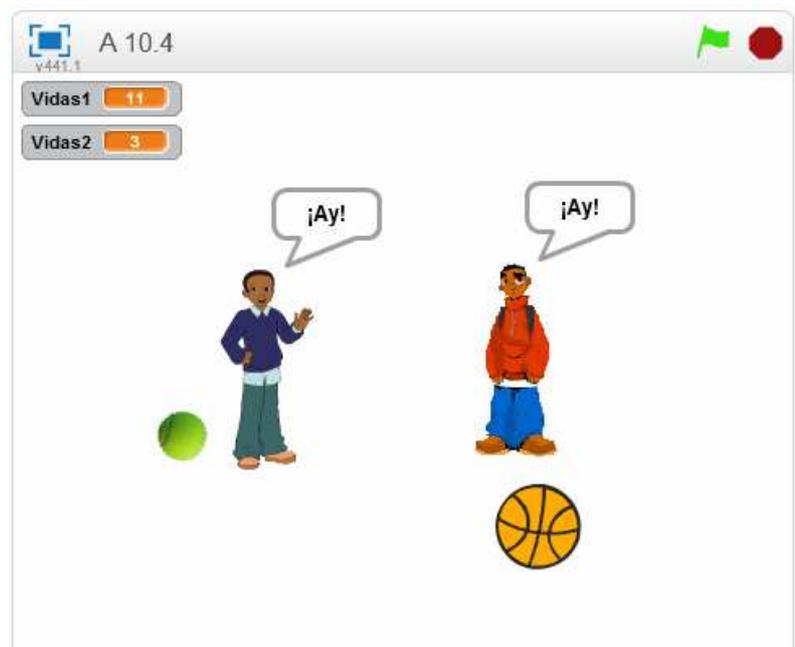
**10.1.-** Hacer que una pelota (biblioteca de Scratch) empiece a moverse en una dirección aleatoria en línea recta, rebotando en los bordes al chocar.

**10.2.-** Haz una variante del anterior programa para que en vez de moverse siempre en línea recta la pelota hasta chocar con bordes, cambie de dirección un ángulo al azar entre  $10^\circ$  y  $20^\circ$  hacia un lado o hacia otro cada 20 pasos.

**10.3-** Hacer que dos objetos empiecen a moverse en direcciones aleatorias y rebotando en los bordes. Si chocan se producirá la explosión y al poco desaparecerán. Entonces aparecerá el gato, que habrá estado oculto, y dirá que las frutas han chocado.

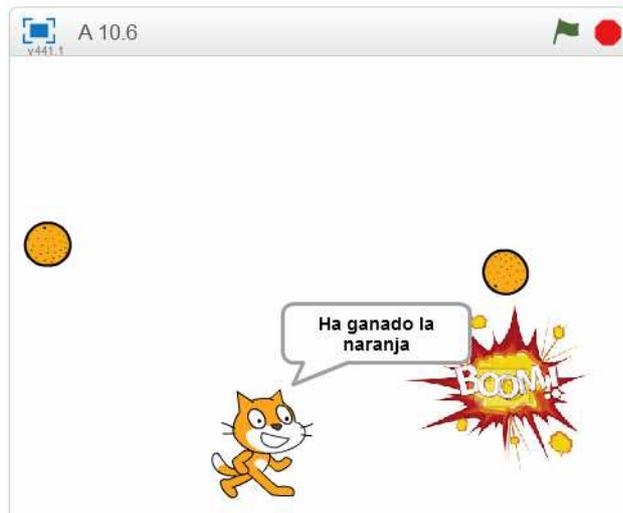
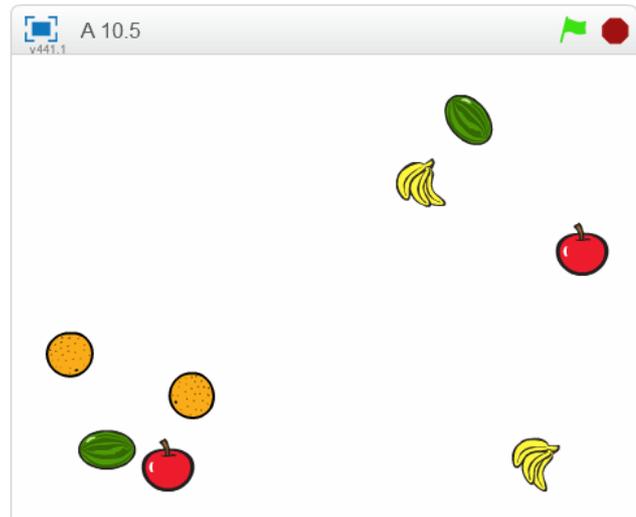


**10.4.-** Un programa que haga aparecer dos personajes en pantalla, quietos. Habrá dos pelotas moviéndose por la pantalla, una verde (la de tenis) y una naranja (la de baloncesto). Cuando la pelota verde choca con un personaje, le añade una vida, mientras que cuando la que choca es la naranja, le quita una vida. Cada personaje empieza con 5 vidas. Cada vez que un personaje recibe un pelotazo dice ¡Ay! El primero que se quede sin vidas pierde. El personaje que acaba con sus vidas, dice el mensaje ¡Adiós mundo cruel! y se volcará.



Para que no se eternice, podemos hacer la pelota naranja más grande que la verde, para que sea más fácil su choque. Cuando las pelotas choquen con los personajes conviene que reboten (girando, por ejemplo, un ángulo al azar entre  $135^\circ$  y  $225^\circ$ ).

**10.5.-** Vamos a aumentar un poco la complejidad de la actividad 10.3. Van a estar moviéndose al azar rebotando en los bordes cuatro parejas de frutas (manzana, melón, banana y naranja). Cuando una fruta choque con su pareja explotarán y desaparecerán. Si choca con una fruta diferente no pasará nada. Así seguirá hasta que sólo quede una pareja de frutas, entonces dejarán de moverse.



**10.6.-** Intenta la siguiente ampliación del programa de la actividad anterior: una vez ya solo se ha quedado una sola pareja de frutas se pararán y además aparecerá el gato y dirá cuál ha sido la fruta ganadora.

Se recomienda utilizar una variable para contar las explosiones y saber cuándo hay que parar.

Para saber la que no ha explotado puedes usar otra variable que vaya cambiando de forma diferente dependiendo de la fruta que explote.

## 11.- Las interacción con el usuario.

**11.1.-** Hacer un proyecto para que podamos mover al gato con las teclas de flechas. Además habrá otro objeto que será el vampiro de la biblioteca. El vampiro se moverá siempre en dirección al gato, usando sus dos disfraces para parecer que vuela. Cuando lo atrape el gato dirá ¡Ayyy! y el vampiro ¡Ñam!



Haz que el juego tenga varios niveles, que varíen la velocidad de movimiento del vampiro (cuanto más rápido va más difícil es esquivarlo). Que al empezar pida el nivel de juego deseado.

**11.2.-** Hacer un proyecto que presente cuatro objetos, uno en cada esquina de la pantalla, y el gato que estará en el centro. Al hacer clic en cualquiera de los objetos el gato se dirigirá hacia él y se parará al tocarlo, hasta que se haga clic en otro objeto.



**11.3.-** Tres bombas se mueven al azar por la pantalla. El gato se mueve con las teclas de flechas. Cada vez que una bomba choca con el gato, le resta una vida de las 7 iniciales. Al chocar la bomba muestra un disfraz tipo ¡Boom! durante un segundo y desaparece, apareciendo en otro punto al azar de la pantalla. Por su parte, el gato dirá un ¡Ay! al ser alcanzado por la bomba. El gato consigue un punto cada vez que consigue atrapar una estrella que habrá en la pantalla. La estrella puede estar fija o moverse al azar rebotando en las paredes. Cada vez que el gato consigue atraparla, la estrella aparece en otro punto al azar en la pantalla. El gato se moverá con las teclas de flechas. Las vidas restantes y los puntos obtenidos aparecerán en pantalla. Ajusta la velocidad de las bombas para que el juego no sea ni muy fácil ni muy difícil.



**11.4.-** Diseñar un juego en el que el gato empieza en la esquina inferior izquierda de la pantalla y debe avanzar mediante las teclas de flechas hasta alcanzar un premio en la esquina inferior derecha. Cuando la alcanza aparece un nuevo premio en la esquina contraria y tendrá que volver a alcanzarlo, y así irá sumando puntos. Sin embargo, en el camino habrá unas pelotas que se mueven verticalmente rebotando, (pueden ir a diferentes velocidades). Cuando una pelota toca al gato le quita una de las vidas iniciales y cambia momentáneamente de disfraz la pelota. Cuando el gato se queda sin vidas acaba el juego y dice ¡Muerto soy! Si juegan varios gana el jugador que haya conseguido más puntos.

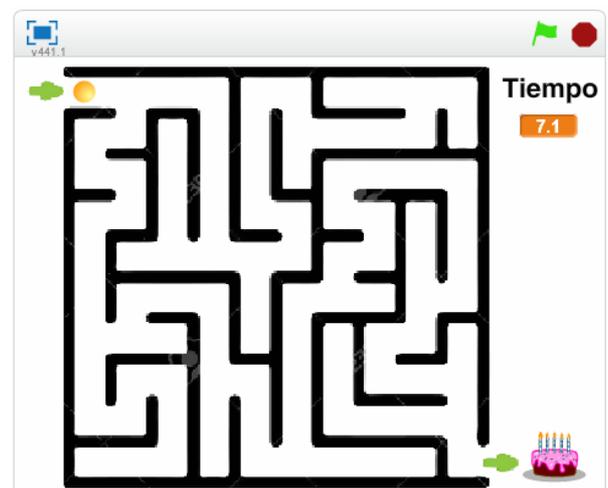


b) Complicar el juego de forma que cuando el gato lleve 10 puntos las pelotas vayan más deprisa. O bien cuando haya pasado un cierto tiempo.

## 12.- El control del tiempo.

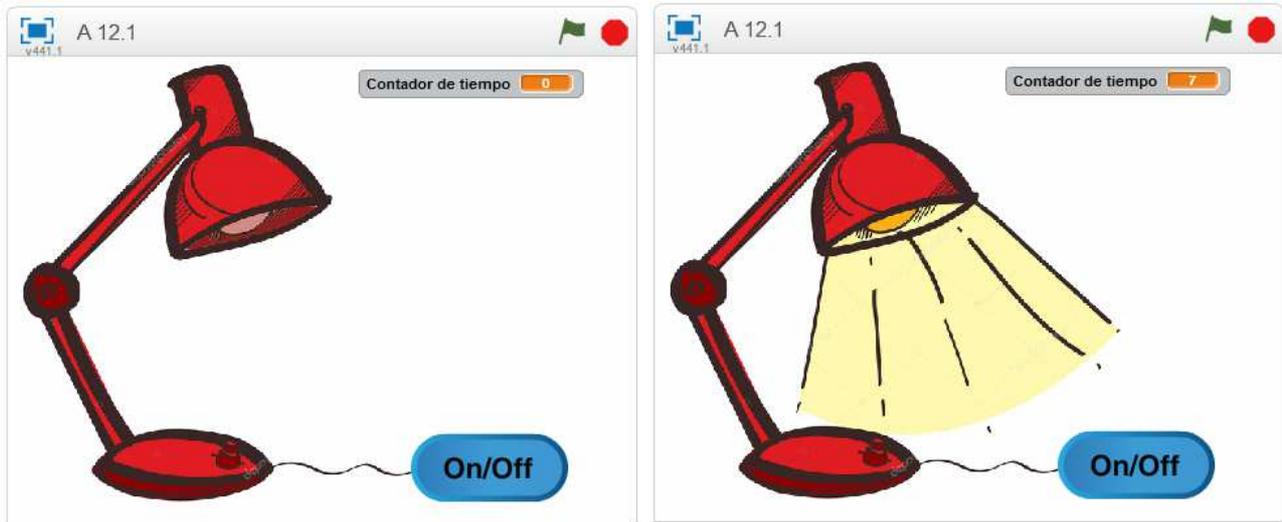
**12.1.-** Con el laberinto que dibujaste en la actividad 1.1 construye un juego para llevar a la bola desde la entrada hasta el pastel que hay a la salida. La bola se moverá con las teclas de flechas. La bola no podrá pasar por las líneas negras. Habrá un contador de tiempo que mida el tiempo desde que se haga clic en la bandera hasta que toca el pastel. El tiempo lo aproximará hasta décimas de segundo.

Nota: conviene que al tocar negro rebote un poco.

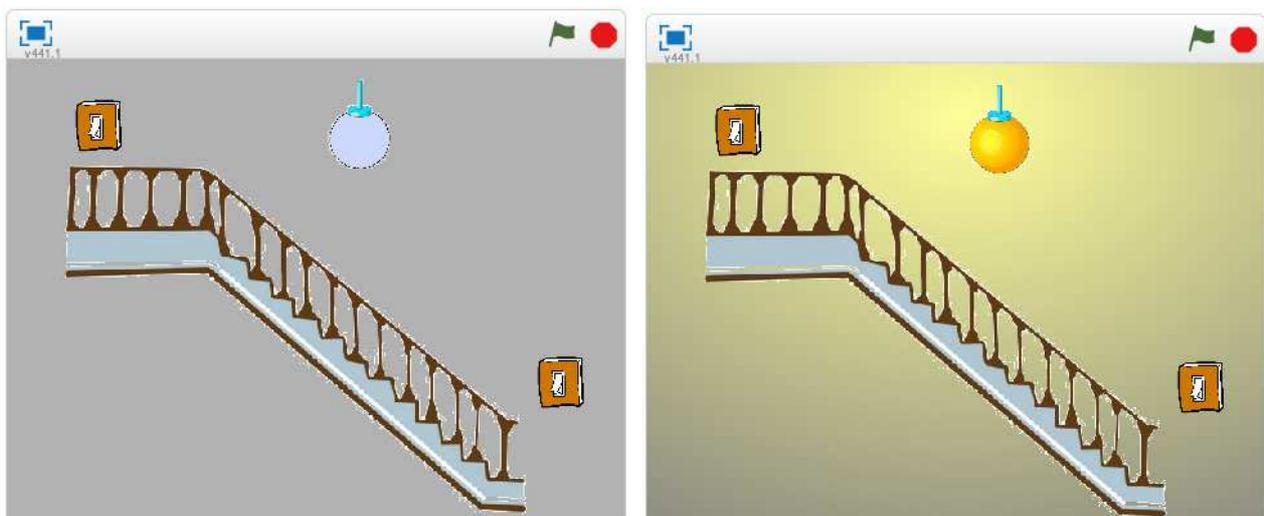


**12.2.-** Hacer un proyecto que conste de los objetos de la figura: un flexo (con dos disfraces, uno encendido y otro apagado) y un botón que vale tanto para encender como para apagar al pulsarlo.

El funcionamiento debe ser el siguiente: al pulsar el botón el flexo cambiará de estado (si está apagado se encenderá y si está encendido se apagará). Ahora bien, si está encendido y no se pulsa para apagarlo en un determinado espacio de tiempo (pondremos 10 segundos para no esperar mucho), el flexo se apagará solo aunque no se pulse el botón. En una esquina de la pantalla aparecerá un contador que irá contando los segundos (no se indicarán las décimas de segundo) que van pasando desde el último encendido. Estará a 0 cuando esté apagado.



**12.3.-** Hacer un proyecto con los elementos de la figura (busca imágenes en internet y manipúlalas con el editor de Scratch) que simula el funcionamiento de una luz de escalera. Cuando la luz está apagada podrá encenderse haciendo clic en el pulsador situado abajo (alguien va a subir) o en el situado arriba (alguien va a bajar). Igualmente, si la luz está encendida podrá apagarse haciendo clic en el pulsador situado abajo o en el situado arriba. En previsión de que alguien se marche olvidando apagar la luz, si pasados 15 segundos de haberla encendido nadie la hubiera apagado, la luz se apagará sola. Observa que el fondo de escenario también cambia.



### 13.- Actividades que combinan ya todos los contenidos

**13.1.-** En este proyecto debemos simular el funcionamiento de las luces automáticas de la entrada del edificio desde la calle y de las dos escaleras (una que permite el acceso a los pisos impares y otra a los pisos pares. Para abrir o cerrar las puertas se hará clic sobre ellas con el ratón. Las puertas cambiarán de disfraz. El fondo del escenario también cambiará dependiendo de las luces encendidas. El funcionamiento será el siguiente:

- Para un vecino que viene de la calle y entra al edificio:

Al abrir la puerta de la calle se encenderá la luz del hall de entrada. Tras cerrar la puerta de la calle hará clic sobre la escalera que da a su piso y entonces se apagará la luz del hall y se encenderá la de la escalera que da a su casa. Dicha luz se apagará cuando el vecino abra y, a continuación, cierre la puerta de su casa.

- Para un vecino que sale de su piso para ir a la calle:

Al abrir la puerta de su casa se encenderá la luz de la escalera correspondiente y la luz del hall de entrada. Cerrará la puerta de su casa. Después, cuando abra la puerta del edificio que da a la calle, se apagará la luz de la escalera, pero permanecerá aún la del hall de entrada. Por último, cuando cierre la puerta de la calle se apagará la luz del hall.

En las figuras se muestran varias situaciones posibles.

Evidentemente, el funcionamiento se ha simplificado para no hacer la actividad muy compleja. Por ejemplo, no se tiene en cuenta que pueden salir varios vecinos a la vez, o que pueden estar saliendo uno y entrando otro, etc. Se supondrá que solo entra o sale un vecino cada vez.

Aunque, si quieres, puedes plantearte como un reto perfeccionarlo contemplando alguna de estas posibilidades.

