APRENDER A PROGRAMAR POR BLOQUES

RETOS PRÁCTICOS Y SUS SOLUCIONES EN SCRATCH

DR. HÉCTOR GALINDO DOMÍNGUEZ

UNIVERSIDAD DEL PAÍS VASCO / EUSKAL HERRIKO UNIBERTSITATEA

APRENDER A PROGRAMAR POR BLOQUES: RETOS PRÁCTICOS Y SUS SOLUCIONES EN SCRATCH

Dr. Héctor Galindo-Domínguez

Departamento de Didáctica y Organización Escolar

Universidad del País Vasco / Euskal Herriko Unibertsitatea

Última versión: abril de 2025

Licencia: CC BY-NC-SA

Material publicado en acceso abierto en la base de datos de Harvard Dataverse: https://doi.org/10.7910/DVN/YEVDOP

ÍNDICE DE RETOS

RETOS BÁSICOS 4
RETOS DE MOVIMIENTO 4
LETRAS GIRANDO 4
UNICORNIO CORRIENDO 4
SISTEMA SOLAR 5
RETOS DE APARIENCIA 6
PINGÜINO PREGUNTÓN 6
BAILANDO EN LA DISCOTECA 6
RETOS DE SONIDO 7
NOTAS DE LA GUITARRA 7
RETOS INTERMEDIOS 8
MOVIMIENTO E INTERACCIONES 8
LA PRINCESA Y EL DUENDE 8
BOLA GIRADORA 9
COPOS DE NIEVE 10
¿MANZANAS O NARANJAS? 11
POLLO SALTARÍN 12
LANZA LA MONEDITA 12
APARCA EL COCHE 13
JUEGOS CLÁSICOS 14
ATRAPA LA MANZANA 14
ARKANOID 16
LUZ ROJA, LUZ VERDE 17
LABERINTO 18
APLICACIONES EDUCATIVAS 19
MAPA DE EUROPA INFORMATIVO 19
DESCUENTO DE PRODUCTOS 20
CALCULADORA DE ÁREAS 20
CONVERTIDOR DE DISTANCIAS 21
CLASIFICACIÓN DE ANIMALES 22
TRADUCTOR COMÚN 23
TRADUCTOR TEXTO A VOZ 24
OTROS 25
VERIFICADOR DE CONTRASEÑAS 25

RETOS AVANZADOS 26
JUEGOS CLÁSICOS 26
EL GANCHO 26
DOODLE JUMP 27
CAZA AL LORO 28
DANCE DANCE REVOLUTION 29
WHAC-A-MOLE 31
FROGGER 32
SIMON DICE 33
SNAKE 34
PING-PONG 35
FLAPPY BIRD 36
APLICACIONES EDUCATIVAS 38
ADIVINA EL NÚMERO 38
SUMA DE DÍGITOS 38
NÚMEROS A SONIDO 39
CÁLCULO MENTAL 40
DIBUJO DE FORMAS GEÓMETRICAS 41
OTROS 42
#LUPA 42
PELOTA REBOTA 42
LISTA DE LA COMPRA 44
CIRCUITO DE CARRERAS 45
CAZANDO AL GATO 46
PIEDRA, PAPEL O TIJERAS 46
RETOS EXPERTOS 47
JUEGOS CLÁSICOS 47
MEMORY 47
PASAPALABRA MINI 48
EL AHORCADO 49
APLICACIONES EDUCATIVAS 50
AGRUPAMIENTO DE GRUPOS 50
OTROS 52
#ENCRIPTADOR/DESENCRIPTADOR 52
DETECTOR DE NÚMEROS PRIMOS 53

APRENDER A PROGRAMAR POR BLOQUES: RETOS PRÁCTICOS Y SUS SOLUCIONES EN SCRATCH

RETOS BÁSICOS

RETOS DE MOVIMIENTO

 # LETRAS GIRANDO

OBJETIVO	
En el escenario se mostrarán dos letras B. Al hacer clic sobre una de ellas, cada vez, esta girará unos grados. Por otra parte, al hacer clic en bandera verde, la otra B estará continuamente girando y el escenario del fondo cambiando automáticamente.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Block-B:** Se introduce desde el repositorio de Scratch. Solo cuenta con un único disfraz. Se deben introducir dos objetos "Block-B". El funcionamiento del primero será: Al hacer clic sobre él, este girará unos grados en el sentido que queramos. El funcionamiento del segundo será: Al hacer clic en bandera verde, este girará automáticamente en el sentido que queramos indefinidamente.	
ESCENARIOS	
• Añade tres fondos de tu gusto. Cuando se presione en bandera verde, el fondo estará continuamente cambiando.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Para evitar que el movimiento sea excesivamente rápido, antes de que comience un nuevo "por siempre" coloca un bloque "esperar … segundos" poniendo una pequeña cantidad de tiempo (ej., 0.1) • Recuerda emplear un bloque "por siempre" en aquellas secciones que quieras que se repitan indefinidamente (ej: cambiar el fondo, o girar continuamente la letra).	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/1VM_V2jIKto	https://bit.ly/3QId10Z

 # UNICORNIO CORRIENDO

OBJETIVO
Al inicio de la animación aparecerá un unicornio en la parte izquierda del escenario. Este se moverá automáticamente hacia la derecha hasta que, al llegar al borde derecho, desaparezca. Mientras está movimiento, el unicornio deberá ir animando el movimiento.
ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS	
• **Unicorn Running:** Se introduce desde el repositorio de Scratch. Cuenta con seis disfraces útiles para animar el movimiento. Al comenzar la animación el unicornio comenzará a moverse de izquierda a derecha, como si estuviese galopando. Al llegar al final, desaparecerá. El jugador, al presionar nuevamente, la bandera verde deberá volver a su posición inicial.	
ESCENARIOS	
• **Jungle2:** Se introduce desde el repositorio de Scratch. Se mostrará continuamente mientras dure la animación.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Si usas un bloque de esconder, recuerda comenzar tu código con un bloque de mostrar. En caso contrario, al comenzar el juego, este mantendrá la última apariencia del objeto. • Si usas un bloque de movimiento, recuerda comenzar tu código con un bloque donde le indique al objeto desde dónde debe comenzar nuevamente. En caso contrario, el objeto comenzará desde la última posición guardada. • Utiliza una condición SI… acompañado de un operador "…>…" para que el juego reconozca si la posición en x es mayor a un valor determinado. En caso afirmativo el unicornio se entenderá que ha llegado al final y se esconderá.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/7Kxek3CUiJ8	https://bit.ly/4h7oCS5

 # SISTEMA SOLAR

OBJETIVO	
Añade un sol en el centro del escenario. Consigue que los distintos planetas giren a distintas velocidades alrededor del sol, sin golpearse el uno con el otro, simulando el sistema solar real.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Sun:** Se introduce desde el repositorio de Scratch. Estará en el centro del escenario al iniciar la animación. • **Earth:** Se introduce desde el repositorio de Scratch. Estará dando vueltas alrededor del sol en tercera posición empezando desde el sol. • **Resto de planetas:** Se deben dibujar o descargar desde internet. Resulta importante descargarlos en formato .png para evitar que mantengan el fondo blanco.	
ESCENARIOS	
• **Stars:** Se introduce desde el repositorio de Scratch. Se mantiene permanente durante todo el tiempo que dura la animación.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Para asegurarse que un objeto gira siempre a una misma distancia de un determinado objeto, emplea un bloque "por siempre". La acción indefinida comenzará mandando al objeto rotatorio a la posición del objeto central y pidiéndole que se mueva una	

determinada cantidad de pasos mientras gira una determinada cantidad de grados (mayor cantidad de grados, más rápido girará).
ADICIONAL
• Cuando el jugador haga clic sobre un determinado planeta, saldrá una ventana que dé información sobre ese planeta. Al presionar en una flecha hacia atrás se cerrará esa ventana y se volverá a observar el sistema solar girando.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/uQ4qoKd1Of4	https://bit.ly/3QNdAGZ

RETOS DE APARIENCIA

 # PINGÜINO PREGUNTÓN

OBJETIVO	
Al comenzar la animación, el pingüino nos saludará. Posteriormente, nos preguntará cómo nos llamamos. En ese momento, el jugador deberá introducir su nombre. Finalmente, el pingüino nos dirá: Hola <nombre del jugador>.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Penguin 2:** Se introduce desde el repositorio de Scratch. Solamente es necesario su primer disfraz. Al presionar bandera verde, nos dirá durante 2 segundo "Hola". Posteriormente, nos preguntará por nuestro nombre. En ese momento, el jugador deberá introducir un nombre. Finalmente, el pingüino nos dirá Hola + el nombre introducido por el jugador.	
ESCENARIOS	
• **Artic:** Se introduce desde el repositorio de Scratch. Estará activo durante la duración de la animación.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Emplea el bloque de sensores "Preguntar… y esperar" para hacerle una pregunta al jugador. • Emplea el bloque de operadores "unir … …" para unir dos partes de texto. • Emplea el bloque de sensores "respuesta" para introducirlo dentro de una parte del bloque "unir … …".	
ADICIONAL	
• Haz que vaya andando por el escenario hasta desaparecer por un lateral. Al iniciar nuevamente la animación deberá estar posicionado en el centro.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/aAsMjEr_Tkw	https://bit.ly/41GCAWH

 # BAILANDO EN LA DISCOTECA

OBJETIVO	
Al iniciar el juego aparecerá Cassy Dance en medio de la discoteca, momento en el que se iniciará la música de fondo. Mientras esté la animación, Cassy Dance estará bailando en el escenario y a la vez el color de las luces de la discoteca irá cambiando continuamente.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Cassy Dance:** Se introduce desde el repositorio de Scratch. Dispone de 4 disfraces útiles para animar el movimiento. Al iniciar la animación, aparecerá en el centro del escenario bailando continuamente.	
ESCENARIOS	
• **Spotlight:** Este escenario se mantendrá activo desde que inicie la animación. El color de las luces irá cambiando paulatinamente de manera automática.	
SONIDOS	**VARIABLES**
• **Dance around:** Se introduce desde el repositorio de Scratch. Este sonido se mantendrá activo desde que inicie la animación.	(No son necesarias)
PISTAS	
• Para cambiar el color al escenario automáticamente puedes hacer uso del bloque “sumar al efecto color”. • Recuerda que en ciertas partes del juego tendrás que usar el bloque de control “por siempre” al ser acciones que no se detendrán (por ejemplo, movimiento de Cassy Dance, o cambiar el color del fondo).	
ADICIONAL	
• Haz que Cassy Dance se mueva por la pista de baile mientras baila. • Añádele pasos de baile: Por ejemplo, que con una tecla pueda saltar, con otra tecla que dé una voltereta…	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/z3ivXxZSwec	https://bit.ly/4irnREB

RETOS DE SONIDO

 # NOTAS DE LA GUITARRA

OBJETIVO
Al iniciar el juego habrá una guitarra y un piano sobre el escenario. El jugador, al hacer clic sobre el piano sonará una nota. De igual modo, se programará mediante el teclado diferentes notas de la guitarra, para que se puedan tocar las notas desde DO hasta SÍ
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS

<table>
<tr><td colspan="2">

- **Keyboard:** Se introduce desde el repositorio de Scratch. Al hacer clic sobre este objeto, se iniciará un sonido elegido por el jugador.
- **Guitar-electric1:** Se introduce desde el repositorio de Scratch. Al presionar distintas teclas del teclado (ej: teclas numéricas 1 a 7), sonará una nota diferente. Por ejemplo: Al presionar el número 1 sonará DO, al presionar el número 2 sonará RE y así sucesivamente.

</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">Spotlight: Este escenario estará continuamente activo mientras dure el juego.</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>

- Los distintos sonidos de la guitarra: C Elec Guitar, D Elec Guitar, E Elec Guitar…
- El sonido elegido para el teclado: C Elec Piano, D Elec Piano, E Elec Piano…

</td><td>(No son necesarias)</td></tr>
<tr><td colspan="2">PISTAS</td></tr>
<tr><td colspan="2">

- En Scratch no se usa la escala comúnmente empleada en España de DO, RE, MÍ… En su lugar se usan letras: C (equivalencia a DO), D (equivalencia a Re)…

</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">

- Haz que el jugador pueda poner o pausar un ritmo de fondo mientras toca el piano o la guitarra.
- Haz una melodía simple que se inicie al presionar bandera verde (puede serte de utilidad emplear los bloques de “esperar … segundos” en caso de tener que hacer silencios.

</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/hHezL3ji-9I</td><td>https://bit.ly/3DcHy3Z</td></tr>
</table>

RETOS INTERMEDIOS

MOVIMIENTO E INTERACCIONES

 # LA PRINCESA Y EL DUENDE

<table>
<tr><td>OBJETIVO</td></tr>
<tr><td>

Al comenzar la animación la princesa y el duende estarán en un lugar situado por la izquierda del escenario. En este lugar comenzarán interaccionarán del siguiente modo:

- Goblin: ¿Dónde vas?
- Princesa: A la montaña
- Goblin: Voy contigo.
- Princesa: Estupendo, vamos los dos juntos.

Posteriormente, ambos irán moviéndose hacia el lado derecho de la pantalla y al llegar al final desaparecerán.

</td></tr>
<tr><td>ELEMENTOS RECOMENDADOS DEL RETO</td></tr>
<tr><td>OBJETOS</td></tr>
<tr><td>

- **Goblin:** Se introduce desde el repositorio de Scratch. Solo es necesario un único disfraz. Al comenzar la animación estará por la parte izquierda de la pantalla, lugar donde iniciará la conversación con la princesa. Posteriormente, se desplazará hasta el lado derecho de la pantalla y desaparecerá.
- **Princess:** Se introduce desde el repositorio de Scratch. Solo es necesario un único disfraz. Al comenzar la animación estará por la parte izquierda de la pantalla, lugar donde iniciará

</td></tr>
</table>

la conversación con el Goblin. Posteriormente, se desplazará hasta el lado derecho de la pantalla y desaparecerá.
ESCENARIOS
• **Mountain:** Se introduce desde el repositorio de Scratch. Estará mostrándose este escenario continuamente mientras dure la animación.

SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)

PISTAS
• Siempre que la posición final de un objeto sea diferente a su posición inicial se recomienda iniciar el algoritmo volviendo a colocarlo en su posición inicial a través del bloque "ir a x … y …". Lo mismo pasa cuando la animación concluye escondiéndolo. Para comenzar el juego se debe iniciar el algoritmo mostrándolo. • En una conversación, mientras uno habla el otro espera. Emplea de manera entrelazada los bloques "decir … durante … segundos" y "esperar … segundos".
ADICIONAL
•

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/cD8y-UVAQ6A	https://bit.ly/3XBq4Fx

 # BOLA GIRADORA

OBJETIVO
Al comenzar la animación tendremos una bola situada detrás de una pista (de atletismo, por ejemplo). Al hacer clic sobre ella nos preguntará cuántas vueltas tendría que dar la bola sobre la pista. La bola dará tantas vueltas como el jugador le haya solicitado. Tras darlas dirá en un mensaje: "ufff qué mareo tras tantas vueltas".
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Beachball:** Se introduce desde el repositorio de Scratch. Al hacer clic sobre ella se le preguntará al jugador cuántas vueltas tendría que dar sobre la pista. Tras introducir el jugador un número, se analizará si este número es mayor a 1 dígito. Si tiene más de 1 dígito el juego dará el mensaje de "El número introducido es muy largo. No podría tirarme todo el día dando vueltas". En caso contrario, la bola dará tantas vueltas como haya solicitado el jugador. Una vez dadas todas las vueltas solicitadas, la pelota se detendrá y dirá el mensaje "Ufff qué mareo tras tantas vueltas".
ESCENARIOS
• **Pista:** Se debe dibujar en Scratch una especie de pista circular donde la pelota tendrá lugar para moverse. Puede ser un círculo relleno de algún color.

SONIDOS	**VARIABLES**
(No son necesarios)	• **Grados Girados:** Variable que cuantifica cuántos grados ha girado ya la pelota del total de grados que debe girar. Cuando

	los grados girados lleguen a los grados totales que debe girar, la pelota se detendrá y dirá el mensaje “ufff que mareo tras tantas vueltas”.
PISTAS	
• Emplea la variable GradosGirados para saber cuántos grados ha girado ya la bola del total que debería girar. Recuerda que una vuelta son 360 grados. • Emplea el bloque de control “repetir hasta que…” para que la bola siga girando hasta que los grados correspondientes a la cantidad de vueltas solicitadas sean igual a los grados que lleva girados la bola.	
ADICIONAL	
• Haz que la pista en vez de ser circular sea como una pista de atletismo de verdad y la pelota siga ese recorrido. Esto supondrá cambios significativos en el código, pues partes del trayecto no serán curvos sino rectos.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/zxed2IqVGIc	https://bit.ly/3DVzHrP

 # COPOS DE NIEVE

OBJETIVO	
Un escenario donde caerán copos de nieve de arriba hacia abajo. Al llegar al suelo desaparecerán.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Snowflake:** Se introduce desde el repositorio de Scratch. Al iniciar el juego se elegirá una posición aleatoria en el eje X y comenzará a caer verticalmente en el eje Y. Al tocar el suelo volverá arriba en una posición aleatoria del eje X y seguirá el procedimiento. Para dar la animación de que está nevando se deben generar más copos de nieve a través del bloque “crear clon de mí mismo”, dándole a cada clon el mismo comportamiento que al copo de nieve original. Cada copo de nieve tendrá un tamaño distinto.	
ESCENARIOS	
• **Winter:** Se introduce desde el repositorio de Scratch. Se mostrará este escenario durante todo el tiempo que dure la animación.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Utiliza el bloque Si …. Si no para saber si el bloque ha tocado el suelo (por ejemplo posición en y < - 175) o si por el contrario aún no lo ha tocado y debe seguir bajando. • Para darle tamaños distintos a los copos de nieve, emplea un bloque “fijar tamaño al …%” unido a un bloque “número aleatorio entre … y …”. • Mientras dure el juego haz que se generé un clon cada pocos segundos. Para evitar la sobrecreación de clones, y sabiendo que se van a generar continuamente cada pocos segundos, se puede indicar que una vez que llegue al suelo, se elimine el clon a través del bloque “eliminar este clon”.	
ADICIONAL	

• Añade alguna canción navideña de fondo. • Añade a la animación distintos elementos relacionados con el invierno (por ejemplo, el objeto snowman andando por el fondo…).	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/kleYECD0jg0	https://bit.ly/41pmKPZ

 # ¿MANZANAS O NARANJAS?

OBJETIVO	
Aparecerá un gato preguntando al jugador si prefiere manzanas o naranjas. Posteriormente, en función de lo que responda el jugador, el gato le dirá que él también ama las [fruta elegida por el jugador] y se acabará el juego. En caso de que el jugador introduzca una respuesta distinta a manzanas o naranjas, se le volverá a hacer la pregunta hasta que responda manzanas o naranjas.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Cat:** Se introduce desde el repositorio de Scratch. Solo se necesita un único disfraz. Al iniciar el juego, el gato preguntará al jugador si prefiere manzanas o naranjas. Posteriormente, el jugador elegirá una de las dos opciones. Seguidamente, el gato dirá que él también ama las [fruta elegida del jugador]. En caso de que el jugador introduzca una respuesta distinta a las dos solicitadas, se volverá a hacer la pregunta hasta responder algo de lo esperado.	
ESCENARIOS	
• **Canyon:** Se introduce desde el repositorio de Scratch. Se mostrará mientras el juego esté activo.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Emplea el bloque "… = …" para saber si la respuesta del jugador es igual a lo esperado. • Emplea el bloque "unir… … " para unir dos partes del texto. En este caso el mensaje del gato + la respuesta que introduzca el jguador. • Utiliza un bloque Si …. Si no…. Para saber si el jugador ha introducido una respuesta esperada o no. En caso de no ser una respuesta esperada se le deberá preguntar continuamente hasta introducir una respuesta esperada. • No olvides que las condiciones que utilices deberán estar activas por siempre, mientras dure el juego.	
ADICIONAL	
• Añade mayor cantidad de preguntas; donde la respuesta del gato en ocasiones sea la contraria a la del jugador. Para conectar una pregunta con la siguiente puedes emplear el bloque "enviar mensaje".	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/wunzPwVRKo4	https://bit.ly/41ekLwq

 # POLLO SALTARÍN

OBJETIVO	
Un pollo que se mueve de izquierda a derecha. Para animar el movimiento, por cada paso que da, cambia de disfraz. De igual modo, cuando presiona la flecha hacia arriba pega un salto, volviendo tras poco tiempo a su lugar inicial del salto.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Chick:** Se introduce desde el repositorio de Scratch. Tiene tres disfraces distintos. Por cada paso que dé se cambiará al siguiente disfraz. El jugador se moverá con las flechas izquierda y derecha. Asimismo, al presionar la barra espaciadora, el pollo saltará hacia arriba y volverá nuevamente al mismo lugar de inicio del salto.	
ESCENARIOS	
• **Blue Sky:** Se introduce desde el repositorio de Scratch. Se mostrará este escenario durante toda la partida.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Para evitar que el pollo pueda aparecer cabeza abajo cuando se mueve hacia uno de los lados, recuerda utilizar el bloque “fijar estilo de rotación” y establecerlo en izquierda-derecha. • Para hacer que el disfraz del pollo cambie de dirección, emplea el bloque “apuntar en dirección” antes de moverte. • Para realizar el salto desglósalo en dos partes: subida del salto y bajada del salto. Para el primero, la cantidad de unidades de Y deben aumentar, mientras que para el segundo, la cantidad de unidades de Y deben reducirse.	
ADICIONAL	
• Presionando una determinada tecla, haz que el pollo pueda volar, pero con el efecto de la gravedad si deja de volar que caiga. • Presionando una determinada tecla, haz que el pollo de una voltereta.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/XwopbBeKxPs	https://bit.ly/4gTC9MI

 # LANZA LA MONEDITA

OBJETIVO
Al hacer clic sobre una moneda se moverá a una posición aleatoria dentro del escenario, a excepción de una zona marcada donde no podrá aparecer. Cada vez que se haga clic sobre ella se moverá aleatoriamente. El escenario estará dividido en dos colores. La moneda si aparece en el área de un determinado color, se convertirá en el color de esa zona; pero si aparece en el área del otro color, se convertirá en el color de esa otra zona. Existirán dos

<table>
<tr><td colspan="2">contadores sobre la cantidad de veces que la moneda ha aparecido en cada una de las dos zonas. El contador se podrá reiniciar al presionar la bandera verde.</td></tr>
<tr><td colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</td></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• Button 1: Se introduce desde Scratch: Debe disponer de dos disfraces, un disfraz del botón pintado de cada color (por ejemplo, uno morado y otro rojo). Al presionar clic sobre el botón, este se moverá a una posición aleatoria dentro del escenario a excepción de la zona prohibida donde no podrá aparecer. Si tras hacer clic sobre él aparece en la zona de color morado, el botón mostrará el disfraz morado; y en cambio, si tras hacer clic sobre él aparece en la zona de color rojo, el botón mostrará el disfraz rojo.</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">• Dibuja un escenario en Scratch. En él se deberá dividir en dos partes iguales, cada una pintada con un color distinto. Asimismo, dentro de cada una, deberá haber una sección prohibida, por ejemplo, reflejada a través de unas líneas diagonales. El botón no podrá generarse en esta área.</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>• Rojos: Variable que permite contabilizar la cantidad de veces que la moneda ha caído en la sección roja. Si se presiona bandera verde el contador se reiniciará.
• Morados: Variable que permite contabilizar la cantidad de veces que la moneda ha caído en la sección morada. Si se presiona bandera verde el contador se reiniciará.</td></tr>
<tr><td colspan="2">PISTAS</td></tr>
<tr><td colspan="2">• No emplees el botón “ir a posición aleatoria”, en cuanto a que habrá unas zonas que estén prohibidas. En su lugar, emplea el bloque “ir a x … y …” y acompáñalo del bloque de operadores “número aleatorio entre … y …”. Recuerda que en el eje X el recorrido es desde X = -240 a X = +240; y en el eje Y el recorrido es desde Y = -180 a Y = +180.</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">• Haz que la moneda del juego se vaya moviendo automáticamente sin necesidad de presionar sobre ella.
• Crea una nueva variable que te indique el porcentaje de veces que la moneda ha caído en un lugar con respecto a la cantidad total de tiradas.</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/wy0-LMKT2CE</td><td>https://bit.ly/3QEP2jk</td></tr>
</table>

 # APARCA EL COCHE

<table>
<tr><td>OBJETIVO</td></tr>
<tr><td>En el escenario aparecerán distintos coches. Al hacer clic en uno, se encenderá el motor, y el jugador con las flechas de movimiento podrá dirigir el coche. En la parte superior, tendrá unos estacionamientos con sensor de aparcamiento: en caso de que el coche esté aparcado sobre</td></tr>
</table>

el hueco de parking, el sensor cambiará a rojo; en caso contrario, al estar el hueco de parking vacío, el sensor cambiará a verde.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Coche:** Se puede introducir desde Scratch, aunque es recomendable descargar desde Google un coche animado con vista superior. Este coche tendrá dos disfraces: un disfraz del coche con una luz verde en algún lado (motor encendido), y otro disfraz del coche con una luz roja en algún lado (motor apagado). El jugador podrá encender o apagar el motor haciendo clic sobre el coche: Si el coche está apagado y hace clic lo encenderá, y viceversa. El coche solo se moverá con el motor encendido. Si el motor está encendido, el jugador podrá usar las flechas de movimiento para desplazarse por la pantalla. Deberá haber un mínimo de 3 coches por el parking.
- **Parking:** Se debe dibujar desde Scratch. Dispone de dos disfraces: un disfraz de la plaza de parking con una luz verde en la parte superior (el parking está disponible) y un disfraz de la plaza de parking con una luz roja en la parte superior (el parking está ocupado). Cuando el coche esté aparcado encima del parking, el sensor de la plaza cambiará a ocupado.

ESCENARIOS

- No es necesario ningún escenario especial, aunque se puede pintar todo de gris como si fuese asfalto.

SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)

PISTAS

- Para generar mayor cantidad de coches con un único objeto emplea el bloque “crear clon de mí mismo”. Recuerda que el código que le programes al coche original, deberás también programárselo a los clones a través del bloque “al comenzar como clon”.
- Para animar el movimiento del coche deberás emplear dobles condiciones, primero para detectar SI el disfraz del coche es el del coche encendido, y posteriormente para detectar SI se está presionando una tecla en concreto. Recuerda que puedes meter una condición dentro de otra condición, o emplear el bloque de operadores “… y …” para meter ambas condiciones en el mismo lado.
- Para mostrar el parking como disponible u ocupado emplea la condición SI… acompañado del bloque “tocando <coche>”.

ADICIONAL

- Prepara el escenario para que solamente se pueda circular por la carretera.
- Enumera las plazas de parking e indícale al jugador qué plazas están cogidas y cuáles no.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/WoMpyx2ygMI	https://bit.ly/4ioHc9f

JUEGOS CLÁSICOS

 # ATRAPA LA MANZANA

OBJETIVO

Una manzana cae paulatinamente desde la parte superior del escenario. La posición inicia en el eje X es aleatoria. Al llegar a la parte inferior del escenario se hallará una plataforma, movida por el jugador de izquierda a derecha con las flechas de movimiento. Si la manzana

toca la plataforma, sumará un punto el juego y la manzana volverá arriba a una posición aleatoria en el eje X. Por el contrario, si la manzana no toca la plataforma, restará un punto el juego y la manzana volverá arriba a una posición aleatoria en el eje X.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Button2:** Se introduce desde el repositorio de Scratch. Solamente necesita un único disfraz. Se posicionará por la parte inferior del escenario. El jugador podrá moverlo de izquierda a derecha con las flechas del teclado.
- **Apple:** Se introduce desde el repositorio de Scratch. Solamente necesita un único disfraz. Al inicio se posicionará en un valor aleatorio del eje X para posteriormente caer verticalmente en el eje Y. Tanto si toca la plataforma del jugador, como si toca el suelo la manzana volverá arriba a una posición aleatoria para volver a caer.

ESCENARIOS

- **Blue Sky:** Se introduce desde el repositorio de Scratch. Mientras dure el juego estará continuamente mostrando este escenario.

SONIDOS	**VARIABLES**
• **Chomp:** Se introduce desde el repositorio de Scratch. Este sonido se escuchará cada vez que la manzana toque la plataforma del jugador.	• **Puntos:** Variable que contabilizará los puntos del jugador. Dará un punto si la manzana toca la plataforma y quitará un punto si la manzana toca el suelo, sin haber tocado la plataforma.

PISTAS

- Para elegir la posición aleatoria de la manzana en el eje X, une el bloque "Ir a x … y …" junto con el bloque "número aleatorio entre … y ….". Recuerda que el eje X en Scratch oscila entre las -240 unidades y las 240 unidades. En este caso, sería recomendable que en el eje X generáse un número al azar entre los comentados.
- Existen diversos modos para hacer que la manzana caiga paulatinamente. Dos propuestas pueden ser: 1.- Emplear el bloque sumar a Y … (lugar donde se deberá introducir un valor negativo). Este bloque añadirá o quitará cada fotograma la cantidad de unidades indicadas al objeto; 2.- Emplear el bloque ir a x … y …" introduciendo en la Y el bloque "… - …" Dentro de este bloque se deberá indicar que a la posición en Y, cada fotograma, se le deberá restar una cantidad específica de unidades.
- Utiliza un sensor "tocando …." Para saber si la manzana está tocando o no la plataforma del jugador.
- Utiliza un operador " … < …" para saber cuándo la manzana ha llegado a "tocar el suelo". Sustituye el primer valor por el bloque "posición en y".

ADICIONAL

- Añade manzanas doradas, que salgan esporádicamente, y que den mayor cantidad de puntos.
- Añade manzanas podridas, que el jugador deberá evitar tocar.
- Añade distintos objetos que proporcionen efectos distintos (por ejemplo, un objeto que acelere la caída de manzana, que caigan varias manzanas a la vez, etc).

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/GPiZFOqZ6mM	https://bit.ly/4be9S2l

 # ARKANOID

<table>
<tr><th colspan="2">OBJETIVO</th></tr>
<tr><td colspan="2">El jugador moverá una plataforma, situada en la parte inferior del escenario, con las flechas izquierda y derecha. En la parte superior habrá distintos bloques. Asimismo, habrá una pelota que se irá moviendo de manera lógica por el escenario. Si la pelota toca uno de los bloques desaparecerá e irá sumando puntos al contador del jugador. Si la pelota toca un borde o la plataforma del jugador, rebotará. En caso de tocar el suelo del juego sin haber tocado la plataforma, el jugador perderá una vida y la pelota volverá a moverse hacia arriba.</td></tr>
<tr><th colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</th></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• Button2: Se introduce desde el repositorio de Scratch. Solo se necesita un único disfraz. Al presionar las teclas izquierda o derecha, la plataforma se moverá acorde a la flecha presionada.
• Bloque: Se puede dibujar u obtener desde internet. Solo se necesita un único disfraz. La pelota al golper el bloque, este desaparecerá y sumará un punto al contador por cada bloque que ha sido golpeado.
• Ball: Se introduce desde el repositorio de Scratch. Solo se necesita un único disfraz. Al iniciar la partida se moverá de manera lógica. En caso de golpear una pared o la plataforma, la pelota rebotará. En caso de tocar el suelo se le quitará una vida al jugador.</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">• Galaxy: Se introduce desde el repositorio de Scratch. Se mostrará mientras dure la partida.
• Party: Se introduce desde el repositorio de Scratch. Se mostrará cuando se hayan roto todos los bloques.
• Stars: Se introduce desde el repositorio de Scratch. Se mostrará cuando el jugador haya perdido todas las vidas.</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>• Vidas: Al comenzar la partida se establecerá en 3. Cada vez que la pelota toque el suelo se perderá una vida. Al llegar a 0 se mostrará el escenario pertinente y el juego se detendrá.
• Bloques: Variable que contabiliza la cantidad de bloques que han sido golpeados. Al llegar a la cantidad máxima de bloques que conforman el escenario, se mostrará el escenario pertinente y el juego se detendrá.</td></tr>
<tr><th colspan="2">PISTAS</th></tr>
<tr><td colspan="2">• Utiliza el bloque “crear clon de mí mismo” para generar tantos clones como bloques quieras que tenga el juego. Este procedimiento tendrás que ir haciéndolo escalonadamente. Por ejemplo: Crear clon de mí mismo > Sumar a x … (repetir X veces).
• Recuerda programar tanto el código para el objeto original como para los clones que generes.
• Utiliza el sensor “tocando <ball>” para sumar puntos cuando la pelota toque el bloque.</td></tr>
<tr><th colspan="2">ADICIONAL</th></tr>
<tr><td colspan="2">• Cambia el comportamiento de los bloques para que en vez de que desaparezcan que una vez sean golpeados, desaparezcan y rebote la pelota.
• Añade bloques de distintas características (por ejemplo, algunos que requieran más veces para romperse, otros que exploten los bloques de los laterales…).</td></tr>
</table>

DEMOSTRACIÓN	PROPUESTA DE SOLUCIÓN
https://youtu.be/jVOjukXFEiA	https://bit.ly/4gVAztQ

 # LUZ ROJA, LUZ VERDE

OBJETIVO	
El mítico juego 1-2-3 Carabin bon ban basado en la serie "El juego del Calamar". El jugador estará en el borde izquierdo del eje Y, mientras que el robot estará en el borde derecho del eje Y. Al comenzar la partida, el robot se dará la vuelta durante unos segundos, el escenario se colocará de color verde, y por ende, el jugador podrá avanzar. Tras unos segundos, el robot se dará la vuelta, el escenario se colocará de color rojo, y, por ende, el jugador deberá quedarse sin moverse en el sitio. Si mientras está el escenario en rojo el jugador se mueve, será atrapado por el robot, le disparará y morirá. Si el jugador logra llevar a la meta, ganará el juego.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Giga Walking:** Se introduce desde el repositorio de Scratch. Dispone de 3 disfraces, útiles para hacer que el personaje se mueve. Es el personaje que moverá el jugador. El jugador, al presionar la flecha hacia delante, podrá moverse únicamente en el eje X, siempre y cuando el escenario sea verde. • **Robot:** Se introduce desde el repositorio de Scratch. Dispone de distintos disfraces, útiles para hacer que está observando y para hacer que está dado la vuelta sin ver. Cuando el escenario sea rojo estará observando, mientras que cuando el escenario sea verde estará dado la vuelta. • **Punto de mira:** Se debe dibujar desde Scratch. Dispone de dos disfraces: uno en negro, que será el que use en las verificaciones, y otro en amarillo que se usará intermitentemente junto al disfraz 1 para señalar al jugador que ha sido atrapado. Cuando el robot se dé la vuelta, permanecerá visualizando que el jugador no se mueve de su sitio. Mientras no se mueva, el punto de mira estará yendo a posiciones aleatorias buscando algo de movimiento. En caso de moverse mientras el robot está observando, el punto de mira irá directo al jugador, y empezará a cambiar intermitentemente entre el disfraz 1 y 2; siendo este un indicador de que el jugador ha sido atrapado.	
ESCENARIOS	
• **LuzVerde:** Escenario pintado en verde. Se marca una raya al inicio desde donde saldrá el jugador, y una raya al final donde deberá llegar el jugador. • **LuzRoja:** Escenario pintado en rojo. Al igual que en el escenario verde, se marca una raya al inicio desde donde saldrá el jugador y una raya al final donde deberá llegar el jugador.	
SONIDOS	**VARIABLES**
OBLIGATORIOS: • **Dun Dun Dunnn:** Se introduce desde el repositorio de Scratch. Sonará cuando estando en luz roja, el jugador sea detectado moviéndose. • **Tada:** Se introduce desde el repositorio de Scratch. Sonará cuando el jugador llegue a la zona de victoria.	• **Luz:** Variable que cogerá dos valores: Verde o Rojo. En función del valor de esta variable el escenario y los jugadores actuarán de un modo u otro. • **Cronómetro:** Variable que permitirá determinar la cantidad de tiempo que tendrá que pasar el robot dado la vuelta así como la cantidad de tiempo que

OPCIONALES: • **Normas:** Se puede grabar un audio, antes de comenzar la partida donde el jugador deberá aceptar las normas del juego. • **Comprobación:** Se puede introducir un audio de un robot mientras el robot está observando si el jugador se mueve o no. • **LuzVerde:** Se puede grabar un audio con la mítica parte de la serie, para que mientras está la luz verde, el robot cante: "Luz verde ya puedes correr y parar".	pasará buscando posibles movimientos una vez esté visualizando. • **ÚltimaPosición:** Variable que recogerá la última posición en X del jugador un fotograma antes de que el escenario cambie a luz roja.
PISTAS	
• La variable ÚltimaPosición será clave para saber cuál fue la posición en X del jugador un milisegundo antes de cambiar de luz verde a luz roja. Mientras se esté haciendo la comprobación (robot visualizando), si la posición del jugador es distinta a la última posición grabada el jugador perderá. • Mientras que se recomienda que el tiempo de la comprobación sea siempre el mismo (por ejemplo, Cronómetro = 3), para el tiempo que está el robot de espaldas, se pueden usar distintos tiempos. Para ello une el bloque "dar a Cronómetro el valor …." Junto con el bloque de operadores "número aleatorio entre … y ….".	
ADICIONAL	
• Añade rayos laser que se activan y desactivan de manera aleatoria tras unos segundos. Esto hará que el jugador pueda avanzar más o menos. • Añade unos objetos que proporcionen ciertas habilidades al jugador (por ejemplo, un escudo, invisibilidad, ir a posición aleatoria…).	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/EH6tB1F7Rtw	https://bit.ly/3X9WeYl

 # LABERINTO

OBJETIVO	
El jugador, al inicio del laberinto tiene una pelota que debe ir movimiento con las flechas de movimiento hasta el final del laberinto. En caso de tocar alguna pared, la pelota se reiniciará y volverá al inicio. Un cronómetro irá contabilizando el tiempo que va tardando el jugador.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Ball:** Se introduce desde el repositorio de Scratch. Solo necesita un único disfraz. Se moverá por la pantalla presionando las flechas de posición del teclado. Si colisiona/toca una línea del laberinto volverá al inicio del juego.	
ESCENARIOS	
• **Laberinto:** Se puede dibujar u obtener desde Internet. Tendrá un inicio y un final. Es recomendable que las paredes del laberinto tengan un color uniforme para que se aplique un sensor a la hora de cuando la pelota toque una pared.	
SONIDOS	**VARIABLES**

• **Disconnect:** Se introduce desde el repositorio de Scratch. Este sonido se escuchará cuando el jugador toque una de las paredes del laberinto. • **Win:** Se introduce desde el repositorio de Scratch. Este sonido se escuchará cuando el jugador llegue al final del laberinto.	• **Tiempo:** Cronómetro que comienza en 0 y aumenta en 1 por cada segundo que pasa. Sirve para medir cuánto tiempo de partida lleva el jugador.
PISTAS	
• Utiliza el sensor "Tocando el color …" para saber cuándo la bola está tocando la pared a través de su color. • Delimita la zona de victoria con un color determinado. Utiliza el sensor "Tocando el color…" para saber cuándo la bola ha llegado a la zona de victoria.	
ADICIONAL	
• Añade objetos que pueda usar el jugador en el laberinto. Por ejemplo, inmunidad durante unos segundos, teletransporte a una posición aleatoria del laberinto…	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/NtVvICiA9UA	https://bit.ly/431B3vi

APLICACIONES EDUCATIVAS

 # MAPA DE EUROPA INFORMATIVO

OBJETIVO	
El objetivo de la presente aplicación es mostrarle al jugador el mapa de Europa. En caso de hacer clic sobre un determinado país de Europa, el juego debe detectar qué país es y proporcionarle información al jugador, como País, Capital y número de habitantes.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Punto de mira:** Para que la selección del país sea precisa, se puede añadir un punto de mira que irá continuamente siguiendo la posición del cursor del ratón. Este punto de mira tendrá en el centro un puntito rojo que servirá como sensor.	
ESCENARIOS	
• El mapa de Europa. Se puede dibujar o añadir desde una foto en internet. Es recomendable que sea en Blanco y negro (para pintarlo en Scratch) o que venga pintado sin nombres.	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Para mostrar la información de un país se deben dar dos situaciones a la vez: que el jugador tenga el ratón presionado y que el color del sensor del punto de mira esté tocando el color del que está pintado el país.	
ADICIONAL	
• Añade ríos o montañas al mapa para proporcionar al jugador información adicional. • Añade el mapa de otros continentes.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**

https://youtu.be/fJguGMvuGls	https://bit.ly/41pFXRJ

 # DESCUENTO DE PRODUCTOS

OBJETIVO	
En la pantalla aparecerá una caja de manzanas. En un primer momento se conocerá el valor de la caja sin descuento. Posteriormente, se conocerá qué descuento se le debe aplicar. Finalmente, se le mostrará al jugador el precio de la caja con el descuento aplicado.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Caja de Manzanas:** Se debe descargar desde internet o dibujarla desde Scratch. Al hacer clic en ella se le preguntará al jugador cuál es el precio de la caja de manzanas. Posteriormente, se le preguntará por cuál es el descuento que se aplica a la caja de manzanas. Finalmente, en la variable "Precio con descuento" se mostrará cuál es el valor de la caja de manzanas tras aplicársele el descuento.	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
(No son necesarios)	• **Precio con descuento:** Precio de la caja de manzajas tras aplicar el descuento. • **Precio sin descuento:** Precio de la caja de manzajas sin aplicar el descuento.
PISTAS	
• Pregunta primero por el precio del kilo de manzanas y almacena el dato en la variable "precio sin descuento". • Pregunta posteriormente por el descuento aplicable. • Emplea los operadores matemáticos de multiplicación y división para calcular cuánto descuento se aplica y cuál es el producto final tras el descuento.	
ADICIONAL	
• Añade distintos productos, viendo cómo el mismo porcentaje de descuento afecta de modo distinto a los diferentes productos.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/A9u482_vwVI	https://bit.ly/3RszuiV

 # CALCULADORA DE ÁREAS

OBJETIVO
En la pantalla se mostrarán cuatro cuerpos geométricos: un rectángulo, un cuadrado, un círculo y un triángulo. Al hacer clic sobre cada uno de ellos, el juego preguntará al jugador por

la información necesaria para finalmente decirle al jugador cuál es el área del cuerpo geométrico seleccionado en base a los parámetros proporcionados por el jugador.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• Distintas figuras geométricas. Se deben dibujar desde Scratch. Cada figura geométrica se introducirá como objeto independiente. Si se opta por 4 figuras geométricas, el proyecto tendrá cuatro objetos.
ESCENARIOS
(No son necesarios)

SONIDOS	**VARIABLES**
(No son necesarios)	• **Respuesta1:** Variable que se puede emplear para almacenar información en caso donde se necesiten dos datos para calcular el área del cuerpo geométrico (por ejemplo, para calcular el área del rectángulo, base y altura).

PISTAS
• Emplea la variable Respuesta1 para almacenar información de la respuesta a la primera pregunta hecha al jugador. Esto será necesario en los casos donde se necesiten dos datos para calcular el área. Por ejemplo, en el caso del rectángulo se comenzará preguntando por la base (se almacena en la variable) y luego se preguntará por la altura (se almacenará en el bloque respuesta). • Emplea los ordenadores matemáticos cuando sea necesario. Por ejemplo, en el caso del triángulo; (base x altura) / 2
ADICIONAL
• Introduce figuras geométricas diferentes como trapecios, trapezoides…

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/HCvP99alUfY	https://bit.ly/4jbOC00

 # CONVERTIDOR DE DISTANCIAS

OBJETIVO
Programa que permite convertir de cualquier medida de longitud (e.g., km, hm, dam, m…) a otra medida de longitud dando su valor equivalente.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Convertir A:** Se dibuja desde Scratch. Pequeño botón que tiene que contener 7 disfraces, uno para cada medida de longitud (km, hm, dam, m, dm, cm, mm). Se entiende que este primer botón será la magnitud en la que se proporcione la respuesta del jugador. • **ConvertirB:** Se dibuja desde Scratch. Idem que con ConvertirB. Se entiende que este segundo botón será la magnitud en la que se transformará la respuesta proporcionada por el jugador.

- **Arrow1:** Se introduce desde Scratch. Introducir 4 de estas: dos para el botón ConvertirA y dos para el botón ConvertirB. Estas flechas permitirán al jugador cambiar las magnitudes de la conversión antes de introducir el valor que quiere convertir. Si presiona la flecha apuntando a la derecha se cambiará el disfraz a una magnitud menor, mientras que si presiona la fecha apuntando a la izquierda se cambiará el disfraz a una magnitud mayor.
- **BotónConvertir:** Botón que se debe dibujar desde Scratch. Tras seleccionar la magnitud de entrada y de salida, el jugador presionará este botón y se le pedirá que introduzca el valor que quiere convertir. Posterior a ello, se le proporcionará la respuesta de su conversión.

ESCENARIOS

(No son necesarios)

SONIDOS	**VARIABLES**
(No son necesarios)	• **Input:** Variable que permite saber cuál es la magnitud de origen que tiene seleccionada el jugador. Coge una de las 7 categorías disponibles (km, hm, dam…). • **Output:** Variable que permite saber cuál es la magnitud a la que se tiene que transformar el valor de origen. Coge una de las 7 categorías disponibles (km, hm, dam…). • **Resultado:** Variable numérica que recoge el valor de la conversión teniendo en cuenta los valores de las variables input y output.

PISTAS

- Para realizar la conversión, tendrás que indicar que operación se va a realizar para cada caso (si toca km a hm, si toca m a mm…). Por ello, es recomendable emplear condiciones dobles (una condición metida dentro de otra condición), a modo que: Si el input es … entonces… En este momento introducir los 7 escenarios posibles del output. Por ejemplo: (Si input = km entonces… (Si output = km entonces…) (Si output = hm entonces…) (Si output = dam…).
- Emplea el operador matemático __*__ para recoger en la variable Resultado cuál sería el valor de la conversión. Por ejemplo, si es Km a m, la variable resultado será Respuesta del jugador * 1000

ADICIONAL

- Permite al jugador que cambie las magnitudes. Introduce magnitudes de volumen, tiempo, capacidad…

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/JYm2-GA6xJQ	https://bit.ly/4j1dveY

 # CLASIFICACIÓN DE ANIMALES

OBJETIVO

Al jugador se le presentará un animal y deberá arrastrarlo en función de si este es vertebrado o invertebrado. Cuando lo haya clasificado, si la clasificación ha sido correcta se le dará un punto. En cualquier caso, correcto o incorrecto, se volverá luego a generar otro animal al azar.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Vertebrados:** Se dibuja desde Scratch. Solo tiene un único disfraz. Se trata de un texto donde aparece escrito "Vertebrados", metido en un rectángulo con el fondo pintado. • **Invertebrados:** Se dibuja desde Scratch. Solo tiene un único disfraz. Se trata de un texto donde aparece escrito "Invertebrados", metido en un rectángulo con el fondo pintado. • **Animales:** Objeto único formado por 10 disfraces. Estos 10 disfraces deben ser animales introducidos desde el repositorio de Scratch, vertebrados e invertebrados. Al comenzar el juego se generará uno de los 10 al azar, y el jugador deberá clasificarlo arrastrándolo hasta el objeto de vertebrados o invertebrados. Tras la clasificación se generará otro animal más al azar. En caso de que la clasificación sea correcta se le dará un punto a los correctos al jugador. En caso de que sea incorrecto, se le dará un punto a los incorrectos.
ESCENARIOS
(No son necesarios)

SONIDOS	**VARIABLES**
(No son necesarios)	• **Correctos:** Variable que cuantifica la cantidad de animales bien clasificados. • **Incorrectos:** Variable que cuantifica la cantidad de animales mal clasificados.

PISTAS
• Utiliza la condición "Sí…" acompañado del sensor "tocando in/vertebrados" y el bloque "número de disfraz" para indicarle al juego qué disfraces pertenecen a animales vertebrados y cuáles invertebrados.
ADICIONAL
• Realiza el mismo juego, pero con diferentes contenidos. Por ejemplo: Estaciones del año, tipos de números (entero, decimal, porcentajes…), tipos de palabras…

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/elmhPBSDn7c	https://bit.ly/4hI5lI7

 # TRADUCTOR COMÚN

OBJETIVO
Un traductor común. Se le dará al jugador un poco de tiempo para que elija el idioma al que quiere traducir el texto. Posteriormente se le preguntará qué texto quiere traducir. Finalmente, teniendo en cuenta el idioma seleccionado, el gato dirá el siguiente mensaje: "El mensaje que has introducido, traducido, se dice: …."
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS

<table>
<tr><td colspan="2">

- **Cat:** Se obtiene desde el repositorio de Scratch. Este personaje será el que pregunte al jugador qué texto quiere traducir así como el texto traducido.
- **Banderas:** Se debe introducir tantas banderas como idiomas se quieran introducir. Cada bandera se colocará como objeto.
- **Tick:** Se debe dibujar u obtener desde internet. Solo tendrá un único disfraz. Servirá para que el jugador sepa en todo momento qué idioma (bandera) tiene seleccionada.

</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>

- **Idioma:** Variable que permite identificar a qué idioma se va a querer traducir el texto que introduzca el usuario.

</td></tr>
<tr><td colspan="2">PISTAS</td></tr>
<tr><td colspan="2">

- Emplea el bloque “Traducir … al …” unido a los bloques “respuesta” y la variable “Idioma” para establecer el texto que se va a traducir.
- Para unir el texto que debe decir el gato junto a la traducción del texto, emplea el bloque “unir … …”

</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">

- Añade una mayor cantidad de idiomas para traducir.
- Introduce una lista donde se recoja en cada elemento la traducción del texto introducido en diferentes idiomas.

</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/MwAL1TbDIsU</td><td>https://bit.ly/41nfRhX</td></tr>
</table>

 # TRADUCTOR TEXTO A VOZ

<table>
<tr><td colspan="2">OBJETIVO</td></tr>
<tr><td colspan="2">Cuando inicie el juego, el jugador tendrá cinco segundos para elegir un idioma de los que se les plantea (por ejemplo, español, alemán y francés). Tras pasar ese breve periodo de tiempo, se fijará el idioma al idioma seleccionado por el jugador y el programa convertirá el texto a audio. Para ello se empleará la extensión “Texto a voz” de Scratch.</td></tr>
<tr><td colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</td></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">

- **Monet:** Se introduce desde el repositorio de Scratch. Solo necesita un único disfraz. Comenzará diciendo al jugador que elija un idioma. Para ello, se le dará al jugador cinco segundos. Posteriormente, preguntará “¿Qué texto debería leer?” Tras introducir el jugador el texto, el juego tendrá en cuenta el idioma que ha seleccionado y leerá con voz el texto introducido.
- **Bandera:** Emplea tantas banderas como idiomas quieras introducir en el programa. Se recomienda introducir un objeto para cada bandera. Para saber qué idioma está seleccionado, se le puede mostrar al jugador un ‘tick’ al lado de la bandera que haya seleccionado. Este tick se irá moviendo en función del idioma que seleccione el jugador.

</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
</table>

(No son necesarios)	• **Idioma:** Variable que irá cambiando en función del idioma seleccionado que tenga el jugador. Permitirá modificar el bloque "fijar idioma a …".
ADICIONAL	
• Añade mayor cantidad de idiomas al programa. • Permite hacer traducciones en el idioma del estudiante + en el idioma seleccionado. Ambas traducciones que se hagan en voz.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/ogcnBanX5is	https://bit.ly/3EKMFZw

OTROS

 # VERIFICADOR DE CONTRASEÑAS

OBJETIVO	
El objetivo de este programa es programar un sistema que detecte que la contraseña que ha introducido el jugador está formada por letras y números. En caso de estar formada solo por letras o solo por números, este dará un error avisándole al jugador por el tipo de contraseña que tiene.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Key:** Se introduce desde el repositorio de Scratch. Al hacer clic en este objeto se le explicará al jugador que tiene que generar una contraseña formada por números y letras. Si la contraseña está formada solo por números o solo por letras, el juego le dirá que esa contraseña no es como la que se está pidiendo y le permitirá introducir nuevamente otra contraseña hasta introducir una contraseña formada por letras y números.	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
(No son necesarios)	• **Verificador:** Variable que permite verificar si una contraseña con LETRAS existe algún número desde 0 al 9. Tomará el valor 0 cuando solo tenga letras, o 1 cuando contenga números.
PISTAS	
• En este reto existen tres posibles escenarios: Que el jugador haya escrito solo números, que el jugador haya escrito solo letras o que el jugador haya escrito letras y números (deseable). Comienza diviendo estas tres opciones en dos: ¿Contiene números o solamente letras (con o sin números)?. Para verificar si un texto es puramente numérico puedes pedirle a scratch que redondee el valor dado. Si es capaz de redondearlo será un número. Si no es capaz de redondearlo es porque será texto (o texto + números). • Dentro de la condición de texto o texto+número deberemos verificar a través de la variable verificador si la respuesta introducida por el usuario contiene algún número desde el 0 al 9. Para ello emplea el bloque de operadores" ¿… está en …?". Si algún número	

desde el 0 al 9 está en la respuesta dada por el usuario entonces la variable verificador será 1. En caso contrario, será 0 y sabremos que el texto del usuario es solamente letras.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/u59EUBQWniw	https://bit.ly/4hWzgvk

RETOS AVANZADOS

JUEGOS CLÁSICOS

 # EL GANCHO

OBJETIVO	
Un gancho colgado en la parte superior se va moviendo de izquierda a derecha. Cuando el jugador presione la barra espaciadora el gancho bajará y agarrará los objetos que haya en la parte inferior. El gancho tendrá una probabilidad del 30% de atrapar el premio. En caso de no dar premio a media subida el gancho soltará el premio. En caso de dar premio llevará el premio hasta el cajetín de premios, lugar en el que se mostrarán y se esconderán posteriormente.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Gancho:** Se debe dibujar desde Scratch. Puede contener bastantes disfraces para animar el movimiento lateral (mover el gancho de izquierda a derecha), así como animar el movimiento de bajada sin enganche y el movimiento de subida con enganche. El gancho iniciará arriba a la izquierda y se moverá con las flechas izquierda y derecha. Al presionar la barra espaciadora, el gancho bajará y agarrará los regalos con los que entre en contacto. En caso de dar premio, arrastrará los premios que lleve hasta la posición inicial y los arrojará al cajetín de premios. • **Premios:** Objeto único con diez disfraces a tu gusto. Se pueden añadir todos desde el repositorio de Scratch: Pelotas, altavoces, comida… Al iniciar el juego aparecerán en la parte superior al cajetín de premios.	
ESCENARIOS	
• **Máquina de gancho:** Se debe dibujar en Scratch la estructura de una máquina de gancho. Para ser más realista se pueden emplear distintas capas para que a la hora de dar premio, la bandeja de premio se abra paulatinamente.	
SONIDOS	**VARIABLES**
• **Sewing Machine:** Se obtiene desde el repositorio de Scratch. Este sonido se escuchará cada vez que el gancho se mueva de izquierda a derecha, así como cada vez que el gancho baje y suba para recoger premio. • **Squeaky Toy:** Se obtiene desde el repositorio de Scratch. Este sonido se escuchará cada vez que el gancho suelte un premio.	• **ProbabilidadAtrapar:** Variable que determinará si tras presionar la barra espaciadora dará premio o no.

• **Tada:** Se obtiene desde el repositorio de Scratch. Este sonido se escuchará cada vez que el gancho proporcione un premio.	
PISTAS	
• Para dar un efecto más realista a la hora del agarre, en el premio, en vez de utilizar el sensor ¿tocando gancho? Se recomienda dibujar en el centro del gancho un puntito rojo, que servirá de sensor de objetos. Por ende, se recomienda en los premios emplear el bloque ¿tocando color rojo? • Emplea el bloque "crear clon de mí mismo" para que aparezca unos 15-20 regalos en la parte donde se deben posicionar los premios. • La probabilidad de atrapar puede venir determinada primero por la generación de un número al azar entre 1 y 10; dando premio si el número es 3 o inferior. • Emplea el bloque "cambiar disfraz a" unido al bloque "número aleatorio entre … y …" para elegir aleatoriamente los premios que aparecerán en el juego. Recuerda que cada disfraz tiene su propio número.	
ADICIONAL	
• Realiza distintas versiones del juego del gancho. Por ejemplo, el gancho de empuje, etc.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/NeH89AWv9TA	https://bit.ly/3X68t8c

 # DOODLE JUMP

OBJETIVO
Al comenzar la partida, habrá distintas plataformas que irán apareciendo de arriba hacia abajo. Estas plataformas serán estables, inestables (se romperán tras un segundo al entrar en contacto tras el jugador) o movibles. El jugador deberá ir aumentando su altura saltando de plataforma en plataforma sin caerse.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Extraterrestre (doodle):** Puede servir cualquier personaje de Scratch, dibujarlo u obtenerlo desde internet. Solo tiene un único disfraz. Este personaje, se podrá mover a la izquierda y a la derecha. De igual modo, cuando presione la barra espaciadora saltará. Si se posiciona encima de un piso se mantendrá encima de él, momento que podrá aprovechar para volver a saltar y continuar su trayectoria hacia arriba. • **Piso:** Se recomienda pintarlo en Scratch. Consta de 4 disfraces: el primero, un rectángulo de color naranja. Estos pisos son estables y el jugador puede mantenerse en ellos sin que provoquen ningún efecto. El segundo, un rectángulo de color azul. Estos pisos son movibles de derecha a izquierda. Cuando llegan completamente a la izquierda se mueven a la derecha completamente. Si el jugador está subiendo en uno de ellos y el piso llega a la izquierda caerá al vacío perdiendo así la partida. En tercer lugar, un rectángulo de color marrón. Estos pisos son inestables. Para esta inestabilidad se usan dos disfraces: el primero, un rectángulo marrón con grietas en el medio; y el segundo, un rectángulo marrón roto completamente. Tras entrar en contacto con el extraterreste, un segundo después el piso marrón se romperá y desaparecerá del juego. Los pisos irán yendo de arriba hacia abajo dando el efecto de que el jugador está subiendo en altura.

- **Piso Inicial:** Piso sobre el que estará apoyado el jugador antes de comenzar la partida. Tras tocar bandera verde comenzarán a bajar pisos. No obstante, el jugador decidirá en qué momento saltar del piso inicial hacia los pisos que están cayendo.

ESCENARIOS

- **Stars:** Este escenario se puede introducir desde el repositorio de Scratch. Se recomienda colocarle dos rectángulos negros a los laterales para que el juego se desarrolle principalmente en formato vertical.

SONIDOS	**VARIABLES**
• **Jump:** Sonido que realizará el extraterrestre por cada salto que realice. • **Lose:** Sonido que se escuchará cuando el extraterrestre caiga al vacío.	• **Tiempo:** Cronómetro que mide la cantidad de tiempo que ha pasado desde el inicio de la partida. Inicia en 0.

PISTAS

- Une el bloque "Cambiar disfraz a…" con el bloque "número aleatorio entre … y …" para que el piso que caiga de arriba abajo sea aleatorio.
- Para añadir el efecto que va a provocar cada piso emplea la condición "Si…" junto al bloque … = …. .En la primera parte deberás colocar el bloque "Número de disfraz". Recuerda que cada disfraz tiene su número.

ADICIONAL

- Añade pisos que provoquen distintos efectos (pisos que suban hacia arriba o hacia abajo, pisos que vayan de izquierda a derecha, pisos que aparezcan y desaparezcan…)
- Introduce objetos que le otorguen ciertas ventajas (o no) al jugador. Por ejemplo, mochila propulsora, etc.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/P4GPJLdTgqg	https://bit.ly/3D7bBtC

 # CAZA AL LORO

OBJETIVO

Aparecerán diferentes loros volando de izquierda a derecha: algunos buenos y otros malos. Ambos tipos se diferenciarán por el color que tienen. El cursor del ratón se transformará en un punto de mira. El objetivo del jugador será disparar a los loros malos evitando disparar a los loros buenos.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Parrot (bueno):** Se añade desde el repositorio de Scratch. Contiene dos disfraces para animar el movimiento del vuelo. Se le añade, además un disfraz adicional para cuando el jugador haya acertado el disparo sobre el loro (por ejemplo, el loro en rojo). El loro se moverá de izquierda a derecha. La cantidad de loros que se generen irá aumentando en función del tiempo de juego que lleve el jugador. De igual modo, la velocidad del loro también aumentará. Si el jugador dispara al loro, este se mostrará en rojo y desaparecerá rápidamente.

<table>
<tr><td colspan="2">

Parrot (malo): Se añade desde el repositorio de Scratch. Contiene dos disfraces para animar el movimiento del vuelo. Se le añade, además un disfraz adicional para cuando el jugador haya acertado el disparo sobre el loro (por ejemplo, el loro en rojo). Se le debe cambiar los colores para diferenciarlo del loro bueno. El loro se moverá de izquierda a derecha. Al igual que en el caso previo, la cantidad de loros que se generen irá aumentando en función del tiempo de juego que lleve el jugador. De igual modo, la velocidad del loro también aumentará. Si el jugador dispara al loro, este se mostrará en rojo y desaparecerá rápidamente.
Punto de Mira: Se debe dibujar en Scratch. Este objeto estará continuamente yendo a donde se sitúe el cursor del ratón.

</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">

Blue Sky: Se obtiene desde el repositorio de Scratch. Se mantendrá este escenario durante todo el periodo del juego.

</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>

Vidas: Cantidad de vidas que tiene el jugador. Comienza en 3 y cada vez que dispara erroneamente al loro bueno pierde una. Al llegar a 0 se detiene el juego.
Puntos: Cantidad de puntos que tiene el jugador. Comienza en 0 y cada vez que disparar correctamente al loro malo obtiene 1.
Tiempo: Reloj interno que va aumentando cada segundo. Mientras mayor sea en tiempo, más rápido irán los loros.

</td></tr>
<tr><td colspan="2">PISTAS</td></tr>
<tr><td colspan="2">

Utiliza el bloque “crear clon de mi mismo” para permitir que en un mismo momento o momentos muy parejos haya más de un loro del mismo tipo en el escenario.
Para quitar vidas o dar puntos debe darse dos situaciones en la misma condición: el loro debe tocar el Punto de Mira y el ratón debe estar presionado.

</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">

Añade objetos que aparezcan aleatoriamente en el escenario y en la duración de la partida. Estos objetos podrán proporcionarle beneficios al jugador (por ejemplo, reducir la velocidad, aumentar las vidas…).

</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/up7iC3eNG4c</td><td>https://bit.ly/41jk9qF</td></tr>
</table>

 # DANCE DANCE REVOLUTION

OBJETIVO
En la pantalla se observará una discoteca con cuatro teclas en la parte superior. Cuando el juego comience el juego tras pulsar la tecla Jolastu, iniciará una canción e irán apareciendo de abajo hacia arriba las mismas teclas que figuran en la parte superior. El jugador deberá

presionar la tecla correspondiente justo en el momento en el que la tecla inferior haya ascendido hasta tocar la tecla superior. Por cada tecla presionada correctamente obtendrá un punto.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Arrow1:** Este objeto debe duplicarse 8 veces. Debe existir 2 objetos con color idéntico de flecha a la izquierda, dos objetos con color idéntico de flecha a la derecha, dos objetos con color idéntico de flecha arriba, y dos objetos con color idéntico de flecha abajo. En total debe existir 8 flechas (8 objetos de este tipo). Una de las dos flechas de cada tipo se debe colocar en la parte superior del escenario, mientras que la otra debe figurar en la parte inferior del escenario y escondida. Cuando el juego inicie, irán paulatinamente saliendo flechas de abajo hacia arriba, en un primer momento con menor frecuencia y posteriormente, cuando haya aumentado la dificultad, con mayor frecuencia. Cada objeto tendrá tres disfraces: uno para cuando está en estado normal, otro para cuando el jugador ha presionado erróneamente la flecha (por ejemplo, antes de tiempo) y otro para cuando el jugador ha presionado correctamente la flecha (cuando ha presionado el botón justo cuando la flecha inferior y superior se tocan).
- **Jolastu:** Se debe dibujar en Scratch. Dispone de 5 disfraces. El primer disfraz es un círculo, funcionando a modo de botón, donde aparecerá "Jugar" o "Jolastu". Una vez presionado en él comenzará una cuenta regresiva de 3 segundos con los siguientes 3 disfraces. Finalmente, el quinto disfraz tendrá escrito "Go" y seguidamente empezará el juego.

ESCENARIOS

Concert: Se obtiene desde el repositorio de Scratch. Durante toda la partida permanecerá este escenario activado.

SONIDOS	**VARIABLES**
• **Canción de Fondo:** Se puede descargar alguna canción de internet e iniciarla una vez haya aparecido el disfraz de "Go" dentro del botón de jugar.	• **Puntos:** Variable que mide la cantidad de teclas que ha presionado correctamente el jugador. Se inicia desde 0. • **Dificultad:** Variable que mide la rapidez con la que suben las flechas en el eje Y. Valores más altos suponen que las flechas suben más rápido, y por ende el juego es más difícil.

PISTAS

- Para controlar la dificultad del juego vincula el bloque "Sumar a y…" la variable "Dificultad". Esta variable aumetará paulatinamente con cada punto obtenido por parte del jugador.
- Para las flechas situadas en la parte superior, emplea dos bloques de condición que se apliquen por siempre. En primer lguar si la tecla inferior está tocando la tecla superior mientras se está presionando la flecha correspondiente, y en segundo lugar si se está presionando la tecla correspondiente, pero la flecha saliente del inferior no está tocando la flecha situada en la parte superior.
- Haz que cada vez que se obtenga un punto, se le sume a la dificultad 0.5 pasos.

ADICIONAL

- Sincroniza alguna canción breve con el comportamiento de las flechas.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/tfejzxHMCto	https://bit.ly/435nlrc

 # WHAC-A-MOLE

<table>
<tr><th colspan="2">OBJETIVO</th></tr>
<tr><td colspan="2">El jugador con el cursor del ratón moverá un martillo por toda la pantalla. Mientras tanto, distintos topos irán apareciendo a través del mapa durante unos segundos. Al de pocos segundos, el topo desaparecerá y aparecerá en otro lado. El objetivo del jugador es golpear al topo con el martillo antes de que desaparezca.</td></tr>
<tr><th colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</th></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• Martillo: Se puede obtener desde internet. Debería tener dos disfraces: uno para cuando no se golpea (más diagonal) y otro para cuando se golpea (más horizontal). Al golpear, se cambiará del primer disfraz al segundo. En todos los demás casos, se usará el primer disfraz.
• Topo: Se recomienda dibujarlo en Scratch. De igual modo, se recomienda que tenga distintos disfraces para generar la animación del topo saliendo de la tierra. También puede resultar interesante tener un disfraz específico que ayudará al jugador a identificar que ha golpeado al topo (por ejemplo, el topo con los ojos con X y triste).</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">• Forest: Se obtiene desde el repositorio de Scratch. Los topos aparecerán y desaparecerán en la zona de césped de este escenario.</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>• Boing: Cada vez que se golpee con el martillo a un topo sonará este sonido.</td><td>• Puntos: Empezará desde 0 y por cada topo golpeado sumará 1 punto.
• VelocidadMostrar: Variable encargada de medir el tiempo que pasa el topo en estado “golpeable”. Mayor cantidad de puntos llevará a una reducción progresiva en el tiempo para ser “golpeable”.
• VelocidadCambioDisfraz: Variable encargada de medir la velocidad con la que se hace la animación desde que el topo deja huella en la tierra hasta que se vuelve a esconder. Mayor cantidad de puntos llevará a una reducción progresiva en el tiempo que se tarda en hacer la animación.</td></tr>
<tr><th colspan="2">PISTAS</th></tr>
<tr><td colspan="2">• Se recomienda que el cabezal del martillo esté continuamente en la misma posición que el puntero del ratón.
• Para saber si el jugador ha golpeado a un topo deben darse tres situaciones: El número del disfraz del topo debe ser el disfraz de “ser golpeable”, el jugador debe tener el puntero del ratón sobre el topo, y además, debe hacer clic del ratón. Une estas tres situaciones a una condición para que el jugador sume puntos.
• Une la variable VelocidadCambioDisfraz/VelocidadMostrar y el bloque esperar … segundos para agilizar más o menos el tiempo que pasa el topo haciendo la animación.
• Emplea el bloque “crear clon de mí mismo” para generar clones del topo. Acompáñalo del bloque “esperar … segundos” para que el juego genere clones del topo cada determinado periodo de tiempo.</td></tr>
</table>

• Recuerda que se debe programar el mismo código para los clones que para el objeto original.
ADICIONAL
• Introduce topos de otro color, que en vez de ayudar a sumar puntos resten puntos si el jugador los golpea.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/j2tmQ2WFKkI	https://bit.ly/4i29hmA

 # FROGGER

OBJETIVO
Un personaje, movido con las flechas de movimiento, debe ir desde la parte inferior del mapa hasta la parte superior. Para ello deberá ir superando ciertos retos como ríos, carreteras, vías de tren… De este modo, deberá buscar el momento oportuno y realizar los movimientos específicos para no caer al río, no ser arrollado por un coche o un tren, o no ser atacado por un erizo.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Trisha:** Se obtiene desde el repositorio de Scratch. Dispone de cuatro disfraces que simulan el movimiento de una persona desde arriba. Este personaje se moverá con las teclas de movimiento en las cuatro direcciones. • **Hedgehog:** Se obtiene desde el repositorio de Scratch. Solo se deben usar los disfraces que simulan el movimiento del erizo. Este personaje se moverá en una zona de césped de izquierda a derecha continuamente. En caso de tocar a Trisha, se perderá la partida. • **Car 1:** Se obtiene desde el repositorio de Scratch. Se recomienda introducir 4 disfraces de coches incluidos en el repositorio de Scratch. Estos coches irán de izquierda a derecha en una carretera. En caso de tocar a Trisha, el juego se detendrá. • **Car 2:** Ídem que Car1. Estos coches irán de derecha a izquierda en la misma carretera que Car1. En caso de tocar a Trisha, el juego se detendrá. • **Madera:** Se dibujará desde Scratch un tronco de madera. Por cada río se tendrá un tronco de madera y el comportamiento de este puede variar. Algunos troncos pueden ir de derecha a izquierda y rebotar, otros en cambio al llegar al borde desaparecer… Si Trisha se sube en un tronco, deberá simular que le está llevando el río. Si se cae del tronco y toca el río, el juego se detendrá. • **Tren:** Se dibujará desde Scratch o se obtendrá de Internet. En las vías de tren, de derecha a izquierda irá un tren cada 4-5 segundos. Este tren irá rápido por lo que antes de arrancar realizará un sonido característico. Si Trisha toca el tren, el juego se detendrá.
ESCENARIOS
• Paisaje: Se debe dibujar distintas áreas horizontales. Algunas áreas pueden incluir: zonas verdes, carreteras de única dirección, carreteras de dos direcciones, ríos, vías de tren… En cada área se hallarán los peligros arriba recogidos en los objetos.

SONIDOS	**VARIABLES**
• **Win:** Cuando Trisha llegue al final del escenario sonará este sonido.	• **Tiempo:** Cronometro regresivo que empezará en 80 e irá descontando

• **Lose:** Todos los objetos que pueden provocar perder la partida (érizo, coche, tren…) activarán este sonido si tocan a Trisha. • **Lose:** Si Trisha toca el río sonará este sonido también. • **Car Horn:** Un poco antes de salir el tren sonará este sonido para avisar al jugador de que abandone las vías de tren.	tiempo. Al llegar a 0 perderá la partida el jugador. • **PosiciónXMadera:** Se deberá crear una variable que recoja la posición en X de cada madera. Esta variable servirá de ayuda para que la posición en X de Trisha siga el recorrido de la madera.
PISTAS	
• Utiliza el bloque "cambiar disfraz a…" y únelo con el bloque "número aleatorio entre … y …" para que continuamente estén saliendo coches diferentes. Observa como cada disfraz tiene un número específico. • Emplea el bloque "crear clon de mí mismo" para que cada X segundos salga un nuevo coche en un sentido hacia otro.	
ADICIONAL	
• Añade distintos niveles al juego.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/aaWJ6p31fto	https://bit.ly/4k7N4oY

 # SIMON DICE

OBJETIVO	
El juego tendrá cuatro botones, uno de cada color y un botón central que permitirá al usuario replicar la secuencia. Una vez el jugador presione el botón central comenzará una secuencia de colores que el jugador deberá recordar. Posteriormente, una vez acabada la secuencia se le preguntará al jugador por la secuencia de colores que ha visto. El jugador deberá escribir la secuencia de colores tal cual ha ocurrido. En caso de acertar, la secuencia de colores será cada vez más larga.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Rojo:** Se debe dibujar en Scratch. Dispone de dos disfraces: uno para mostrar cuando el botón rojo no esté presionado y otro para simular que está presionado. • **Verde:** Ídem que en el caso del botón rojo. • **Azul:** Ídem que previamente. • **Amarillo:** Ídem que previamente. • **Botón central:** Se debe dibujar en Scratch. Cuando se presiona permite al jugador iniciar una nueva secuencia de colores.	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
• **Basketball Bouce:** Sonido que se escuchará cuando el botón rojo se presione. • **Boing:** Sonido que se escuchará cuando el botón azul se presione.	• **Solución:** Variable numérica que asocia cada color-botón a un número. Sirve para generar al azar un número entre el 1 y el

• **Jump:** Sonido que se escuchará cuando el botón verde se presione. • **Squish pop:** Sonido que se escuchará cuando el botón amarillo se presione.	4. En función del número que salga se activará un botón de color u otro. • **Puntos:** Variable numérica que comenzará en 0 e irá en aumento con cada acierto. • **Dificultad:** Variable numérica que indica la cantidad de colores que formarán una secuencia. Cuando la dificultad es 2(mínimo), la secuencia estará formada por 2 botones; Si la dificultad es 5 (máximo) la secuencia estará formada por 5 botones. • **Emaitza:** Lista que transforma la variable Solución (numérica) en texto.
PISTAS	
• Emplea el bloque "Repetir …" para indicarle al juego la cantidad de botones de colores que debe formar la secuencia. Dentro del repetir, la cantidad de botones vendrá determinado por la variable Dificultad. • La variable Solución será la que determine qué botón se debe ir presionando con cada número aleatorio. Emplea 4 condiciones, una para cada color. Si Solución = 1…; Si solución = 2…. • Utiliza el bloque "Enviar mensaje" para vincular objetos entre sí.	
ADICIONAL	
• En vez del usuario tener que escribir la secuencia de colores, puede tocar los botones en el orden correspondiente. • Se puede personalizar. En vez de utilizar colores, se pueden añadir partes del cuerpo, instrumentos musicales…	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/lqZfGk1pCDs	https://bit.ly/4i6JBFB

 # SNAKE

OBJETIVO
Una serpiente se va moviendo ininterrumpidamente por el mapa. El usuario al presionar las flechas de posición cambiará su dirección. El objetivo será ir comiéndose manzanas que aparecerán aleatoriamente por el mapa. Al comer las manzanas la serpiente se irá haciendo cada vez más grande. Si toca un borde, el jugador perderá.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Cabeza:** Se dibuja desde Scratch. Solo se requiere de un único disfraz. Se trata de una cabeza con dos ojos. Será la cabeza de la serpiente. Puede ser un círculo uniforme con dos puntos negros. Al presionar las teclas de movimiento cambiará la dirección de hacia dónde se mueve la serpiente. • **Cola:** Se dibuja en Scratch. Solo se requiere de un único disfraz. Se trata del mismo diseño que la cabeza, pero sin ojos. Su longitud dependerá de la cantidad de manzanas comidas.

• **Apple:** Se obtiene desde el repositorio de Scratch. Solo se requiere de un único disfraz. Si la cabeza toca la manzana irá a otra posición aleatoria del mapa, sumará un punto y además la longitud y velocidad de la serpiente aumentarán un poco.
ESCENARIOS
(No son necesarios)

SONIDOS	**VARIABLES**
• **Lose:** Se añade desde el repositorio de Scratch. Este sonido iniciará cuando el jugador pierda la partida, que será cuando la serpiente toque un borde.	• **Longitud:** Variable que determina la longitud de la serpiente. Irá en función de la cantidad de manzanas comidas. • **Puntos:** Variable que comenzará desde 0 e irá aumentando en 1 por cada manzana comida. • **Velocidad:** Cantidad de pasos que moverá la cabeza. Mientras más puntos, la velocidad también irá aumentando.

PISTAS
• La longitud de la cola se recomiendo realizarla a través de la generación de clones. Para ello usa el bloque “Por siempre” y “Crear clon de mí mismo”. • Los clones deberán ir unidos a la cabeza. Para ello, emplea el bloque “Ir a <Cabeza> y esperar un poco de tiempo para que el clon lleve la demora de la posición que tuvo la cabeza.
ADICIONAL
• Hacer que aparezcan manzanas doradas que den más puntos. • Hacer que cuando el cuerpo choque con la cabeza el jugador pierda.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/nZolf6QYBE4	https://bit.ly/430Nojc

 # PING-PONG

OBJETIVO
Se debe programar un juego de ping-pong donde el jugador juegue contra el ordenador. El ordenador siempre deberá ganar la partida. Tanto el jugador como el ordenador tendrán un portero (plataforma) que se podrá mover de arriba abajo. En el caso del ordenador se moverá desde el inicio de la partida automáticamente.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Ball:** Se introduce desde el repositorio de Scratch. Solo es necesario un único disfraz. Cuando comience la partida la pelota estará continuamente moviéndose por el escenario. En caso de tocar una portería se le añadirá un punto al jugador contrario. En caso de tocar un borde o un portero, en ambos casos, rebotará. • **Porteria 1:** Se debe dibujar en Scratch. Solo es necesario un único disfraz. Constará de una raya vertical de arriba a abajo que será la portería del jugador. Se recomienda pintarla de un color determinado.

- **Portería 2:** Se debe dibujar en Scratch. Solo es necesario un único disfraz. Constará de una raya vertical de arriba abajo que será la portería del ordenador. Se recomienda pintarla de otro color determinado.
- **Button2:** Se puede obtener desde el repositorio de Scratch. Solo es necesario un único disfraz. Hará la función de portero del jugador. Se moverá de arriba abajo con las teclas del teclado.
- **Button 2:** Se puede obtener desde el repositorio de Scratch. Solo es necesario un único disfraz. Hará la función de portero del ordenador. El ordenador lo moverá automáticamente para que no se le pueda meter gol jamás.

ESCENARIOS

(No son necesarios)

SONIDOS	**VARIABLES**
(No son necesarios)	• **PuntosPC:** Puntos que va acumulando el ordenador con cada gol que nos marca. • **MisPuntos:** Puntos que voy acumulando con cada gol que marco. Siempre será 0 porque al PC no se le podrá marcar gol. • **PosicionenYPelota:** Variable clave para poder mover automáticamente el portero del PC. Estará continuamente calculando la posición en Y de la pelota.

PISTAS

- El portero del ordenador se moverá automáticamente siempre a la posición en Y de la pelota. Por ende, la variable PosiciónenYPelota es clave.
- Emplea el bloque "Apuntar en dirección" para cambiar la dirección de la pelota una vez que toque al portero.
- Para evitar que cuando choque contra un portero siempre vaya en línea recta la pelota se recomienda emplear el operador "número aleatorio entre…". Si la pelota va en dirección negativa, el rebote deberá ser positivo (por ejemplo, número al azar entre 70 y 110). En cambio, si va en dirección positiva, el rebote deberá ser negativo (por ejemplo, número -70 a -110).

ADICIONAL

- Con el paso del tiempo se puede programar para que la velocidad de la pelota aumente.
- Es posible programar que exista la posibilidad de que al ordenador se le meta gol.
- Es posible programar un ping-pong de 4 jugadores (con juego automático o no).

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/JEPE51r0Q5s	https://bit.ly/3Qmcu4L

 # FLAPPY BIRD

OBJETIVO

Se presenta un pájaro que de manera estática va moviendo las alas. Si se presiona la barra espaciadora subirá poco a poco, mientras que si no presiona nada bajará. En el transcurso de la partida habrá una serie de tuberías de distintos tamaños que irán apareciendo en temporalidades diferentes yendo de derecha a izquierda simulando el movimiento del pájaro.

El objetivo será conseguir la mayor cantidad de puntos sin tocar el suelo ni las tuberías. A medida que se obtenga mayor cantidad de puntos, la cantidad de tuberías que aparecerán también aumentará.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Toucan:** Se obtiene desde el repositorio de Scratch. Solamente se necesitan los disfraces toucan-b y toucan-c. A estos disfraces se le puede añadir un nuevo disfraz del toucan pintado en rojo, para señalar al jugador que ha tocado el suelo o ha tocado una tubería, perdiendo por ende una vida.
- **Tubería inferior:** Se debe dibujar en Scratch. Este objeto debe tener tres disfraces; cada disfraz siendo una tubería de tamaño distinto.
- **Tubería superior:** Se debe dibujar en Scratch. Al igual que previamente, este objeto debe tener tres disfraces; cada disfraz siendo una tubería de tamaño distinto.

ESCENARIOS

(No son necesarios)

SONIDOS	**VARIABLES**
• **Goose:** Sonido que debe reproducir el tucán cuando pierda 1 vida, ya sea por tocar el suelo o por tocar una tubería.	• Vidas: Contabilizará la cantidad de vidas que le quedan al jugador. El jugador comenzará con 3 vidas y al llegar a 0 perderá. Se perderán puntos si se toca el suelo (por cada segundo tocando el suelo 1 vida), o por tocar una tubería. • Puntos: Cada vez que el tucán pase a través de una tubería en el eje Y sin tocarla sumará 1 punto al marcador. • Dificultad: Corresponderá a los segundos que se debe esperar entre la generación de una tubería y otra. Un número más pequeño supondrá que se generarán tuberías más de seguido, aumentando así la dificultad del juego. La dificultad aumentará en base a la cantidad de puntos que tenga el jugador.

PISTAS

- Utiliza el bloque “crear clon de mí mismo” para permitir que al mismo tiempo haya más de una tubería en la pantalla.
- Por el efecto de la gravedad, el pájaro siempre estará perdiendo unidades de Y mientras no se pulse la barra espaciadora, momento en el que subirá un poco.
- Une los bloques “cambiar disfraz a” junto a “número aleatorio entre … y …” para conseguir que el tamaño de la tubería sea continuamente aleatoria.

ADICIONAL

- Añade tuberías especiales que suban y bajen.
- Añade pájaros explosivos que vayan de derecha a izquierda en línea recta, y que el pájaro deberá evitar.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/xI2eVvP3YkI	https://bit.ly/4k09gBn

APLICACIONES EDUCATIVAS

 # ADIVINA EL NÚMERO

<table>
<tr><th colspan="2">OBJETIVO</th></tr>
<tr><td colspan="2">El mago pensará un número aleatorio entre el 1 y el 100. El jugador irá tratando de adivinar de qué número se trata. Para ello, tras introducir un número erróneo, el mago irá dando pistas sobre si el número que está buscando es mayor o menor al introducido por el usuario. En caso de fallar se le dará infinitas oportunidades hasta adivinar el número.</td></tr>
<tr><th colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</th></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• Wizard: Se obtiene desde el repositorio de Scratch. Solo se necesita un único disfraz. El mago pensará en un valor aleatorio entre 0 y 100. Posteriormente preguntará al usuario para que adivine el número. El jugador irá introduciendo en cada ocasión un número. En caso de acertar el juego finalizará con un mensaje del mago indicando que se ha adivinado. En caso de fallar, el mago dará pistas sobre si el número introducido es mayor o menor al esperado. En caso de fallo se le permitirá al jugador indefinidamente volver a introducir otro valor hasta acertar el valor esperado.</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>• NúmeroMago: Valor que elegirá el mago aleatoriamente entre el 0 y el 100 al comenzar el juego.</td></tr>
<tr><th colspan="2">PISTAS</th></tr>
<tr><td colspan="2">• Utiliza tres condiciones para determinar el comportamiento del mago en función de si el númeroes menor, mayor o igual al buscado.
• Utiliza el bloque “por siempre” para que el juego esté continuamente comprobando las tres condiciones previas.</td></tr>
<tr><th colspan="2">ADICIONAL</th></tr>
<tr><td colspan="2">• En vez de si el número es mayor o menor, el mago puede dar pistas sobre si el número comienza con el primer número de la respuesta dada por el usuario.</td></tr>
<tr><th>DEMOSTRACIÓN</th><th>PROPUESTA DE SOLUCIÓN</th></tr>
<tr><td>https://youtu.be/sdp8R36OdEw</td><td>https://bit.ly/4b7MkfI</td></tr>
</table>

 # SUMA DE DÍGITOS

<table>
<tr><th>OBJETIVO</th></tr>
<tr><td>Al iniciar el juego aparece un ordenador en medio del escenario. El jugador, al hacer clic en él, se le pedirá que introduzca un número. El juego hará la suma de todos los dígitos que compongan ese número y devolverá un mensaje al jugador con la respuesta de tal suma.</td></tr>
<tr><th>ELEMENTOS RECOMENDADOS DEL RETO</th></tr>
<tr><td>OBJETOS</td></tr>
</table>

- **Laptop:** Se introduce desde el repositorio de Scratch. Al hacer clic en él, se le preguntará al jugador por un número de la longitud que quiera. Tras enviar la respuesta, el juego devolverá al jugador un mensaje con la suma de todas las cifras del número introducido.

ESCENARIOS

- **Space:** Se introduce desde el repositorio de Scratch. Elemento puramente decorativo.

SONIDOS	**VARIABLES**
(No son necesarios)	• **Nº números:** Variable que permite ir conociendo la posición de los números, uno a uno, de la respuesta introducida por el jugador. • **Suma:** Variable que va sumando al resultado (al comenzar el juego, 0) el valor del próximo número de la respuesta del jugador. • **Números:** Lista que desglosa en orden los números de la respuesta del jugador. Cada elemento de la lista, en orden ascendiente, corresponde a un dígito de la respuesta del jugador.

PISTAS

- Una vez recogida la respuesta del jugador, debes conseguir desglosar los dígitos uno a uno en la lista "Números".
- En la variable "Suma" vete guardando el resultado de la suma y añádele el valor del próximo dígito en la lista.

ADICIONAL

- Realiza el mismo procedimiento, pero multiplicando los dígitos introducidos en la respuesta del jugador.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/3Gq-pWSvXxM	https://bit.ly/4ln5WkL

 # NÚMEROS A SONIDO

OBJETIVO

Al comenzar el juego se le pedirá al jugador que introduzca una secuencia de números. Cada número irá asignado a una letra. Por ejemplo, DO = 1, RE = 2… así como el 0 será un silencio. Tras introducir el número, el programa reproducirá la canción solicitada.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Speaker:** Se introduce desde el repositorio de Scratch. Este objeto almacenará todos los sonidos de un determinado instrumento, desde el Do al Do Mayor. En función del número que toque en un determinado momento, sonará uno u otro sonido.

ESCENARIOS	
• **Concert:** Se introduce desde el repositorio de Scratch. Elemento meramente decorativo.	
SONIDOS	**VARIABLES**
• Desde el repositorio de Scratch se debe introducir toda una octava de un instrumento. Por ejemplo, C Sax, D Sax, E Sax… C2 Sax.	• **Posición:** Variable que permite ir introduciendo en la lista *Números* la secuencia de números uno a uno proporcionados en la respuesta del jugador. • **Números:** Lista que coloca por orden de introducción la secuencia de números. Cada elemento de la lista corresponderá a un número de la respuesta del jugador.
PISTAS	
• Emplea el bloque "repetir" junto al operador "longitud de" para ir introduciendo uno a uno los números de la respuesta del jugador en la lista. • Cada vez que suene un sonido, puede resultarte interesante eliminarlo de la lista. De este modo, solamente tendrás que establecer una condición que mire si el elemento 1 de la lista es igual a un determinado número.	
ADICIONAL	
• Añade una condición que le indique al jugador si algún carácter de su respuesta es incorrecta (por ejemplo, si en vez de números, ha introducido letras o caracteres no válidos como *, %, $...).	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/KRn_8JQEuzc	https://bit.ly/3FEnh8n

 # CÁLCULO MENTAL

OBJETIVO	
El objetivo es realizar una serie de operaciones de cálculo mental para que al final el usuario introduzca un resultado y ver si va acorde a la respuesta calculada. En caso de no ir acorde a la respuesta esperada, el usuario podrá volver a introducir un número hasta que acierte.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Cat:** Se obtiene desde el repositorio de Scratch. Solo se necesita un único disfraz. Al hacer clic sobre la bandera verde, empezará a realizar una serie de operaciones matemáticas aleatorias, y al último resultado, le añadirá otra operación. Este procedimiento se repetirá varias veces hasta que finalmente el usuario pueda introducir la respuesta. Por ejemplo, ¿cuánto es 5 + 4? (da unos segundos) ¿… + 4? (da unos segundos) ¿… - 6?. Respuesta:	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
(No son necesarios)	• **Símbolo:** Lista que contiene los símbolos que pueden aparecer en las operaciones. Se recomienda que solo incluya 3: +, - y x.

	Las divisiones pueden dar problemas si se hacen con decimales. • **Elemento:** Traslada el símbolo obtenido en el listado previo a esta variable. • **Valor 1:** Primer valor de la operación. En operaciones diferentes a la primera, puede almacenar el resultado de las operaciones previas. • **Valor 2:** Segundo valor de la operación. En cada operación irá cambiando por un valor aleatorio.
PISTAS	
• Genera por una parte la primera operación, y posteriormente el resto de operaciones. • Utiliza una condición con su contra-condición a través del bloque "Si… si no…" para que si la respuesta es correcta finalice el juego y sino se le permita al jugador continuar probando números.	
ADICIONAL	
• Diseña el juego para que admita realizar divisiones exactas.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/vig_IyxtwKE	https://bit.ly/41hOg1A

 # DIBUJO DE FORMAS GEÓMETRICAS

OBJETIVO	
Con la herramienta "Lapiz" de Scratch (Añadir Extensión > Lápiz) se deberá dibujar automáticamente un cuadrado y un triángulo.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Ball:** Se obtiene desde el repositorio de Scratch. Solamente tiene un único disfraz. Este objeto, al hacer clic sobre él se encargará de borrar todo lo dibujado previamente. Al presionar una determinada tecla dibujará un cuadrado y al presionar otra determinada tecla dibujará un triángulo.	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
(No son necesarios)	(No son necesarias)
PISTAS	
• Utiliza la extensión "Lápiz" de Scratch. Combínala con los bloques "Bajar lápiz" y "subir lápiz para pintar cuando necesites. • Utiliza el bloque de movimiento "mover … pasos" para dibujar mientras el lapiz está bajado. • Piensa en cuántos grados debe girar el objeto con el que estás pintando. Por ejemplo, en el cuadrado, cada giró será de 90 grados (necesitarás 4 ángulos rectos).	
ADICIONAL	

• Dibuja otras figuras geométricas, como rectángulos, o círculos. • Dibuja formas simples, como una casa.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/gNGG30Ezi1Y	https://bit.ly/3CGp8sc

OTROS

#LUPA

OBJETIVO	
Te moverás con una lupa por el escenario. Habrá un gato en miniatura que, al acercársele la lupa, se amplificará su tamaño, haciendo la función de una lupa.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Cat:** Se obtiene desde el repositorio de Scratch. Solo se necesita un único disfraz. • **Lupa:** Se puede dibujar u obtener desde internet. Contiene un único disfraz. Es un objeto meramente decorativo que irá continuamente vinculado al cristal de la lupa. • **Cristal de la lupa:** Se recomienda dibujarlo en Scratch. Contiene un único disfraz. Este objeto, en cuanto entre en contacto con algún otro objeto del escenario, irá progresivamente aumentando su tamaño. En contra, cuando no esté en contacto, irá reduciendo progresivamente su tamaño.	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
(No son necesarios)	• **Zoom (Recomendado):** Variable que permite aumentar o reducir progresivamente el tamaño de los objetos.
PISTAS	
• Emplea la variable zoom para que el aumento del objeto sea progresivo y no insofacto. • Emplea el bloque “fijar tamaño” unido a la variable Zoom para que el tamaño se vaya fijando progresivamente cuando el cristal entra en contacto con el objeto.	
ADICIONAL	
• Además de amplificar objetos, se puede probar a amplificar texto.	
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/eJJiFlw88Pk	https://bit.ly/4i2KGho

 # PELOTA REBOTA

OBJETIVO

Se trata de conseguir animar con cierto realismo las físicas de una pelota que se lanza desde una altura determinada. El jugador, antes de lanzar la pelota, podrá colocar la pelota donde desee. Con cada rebote, se le aplicará un coeficiente de restitución (energía que pierde la pelota al golpear contra el suelo). Cuando el coeficiente de restitución es 1, la bola no perderá energía y siempre llegará a la misma altura. En contra, cuando el coeficiente es 0, la bola perderá completamente la energía y no rebotará. Para conocer el coeficiente de restitución que quiere aplicar el jugador se le preguntará al jugador cuál será este coeficiente.
Además, para mejorar el realismo se deberá tener en cuenta el efecto de la gravedad, haciendo que justo cuando esté más arriba la pelota caiga más lento, pero mientras más abajo más rápido.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Basketball:** Se introduce desde el repositorio de Scratch. El jugador podrá mover la pelota a cualquier altura. Al presionar la barra espaciadora la pelota caerá desde la posición que esté. De este modo, simulará el rebote de la pelota, donde irá perdiendo energía con cada rebote hasta finalmente detenerse. De igual modo, la velocidad de caída NO será constante, recogiendo la influencia de la gravedad (un tipo de aceleración) en la que mientras más cerca esté del salto máximo más lento irá, mientras que cuando más cerca del suelo más rápido irá.
- **Suelo:** Se debe dibujar en Scratch. Consta de una línea negra de izquierda a derecha situada aproximadamente en la altura y = -25 (se sitúa un poco más abajo del ecuador porque los cálculos con la pelota se harán a y = 0, pues es lo más fácil. El suelo, por ende, estará un poco más debajo de la pelota).

ESCENARIOS

(No son necesarios)

SONIDOS	**VARIABLES**
• **Basketball bounce:** Sonido que se escuchará cada vez que la pelota toque el suelo.	• **Gravedad:** Variable que permite simular la aceleración de la pelota cuanto más tiempo lleva cayendo. • **AlturasMáximas (opcional):** Lista que permite, antes de que caiga la pelota, registrar la altura máxima de cada rebote aplicándole el coeficiente de restitución introducido por el jugador. Cuando llegue a 0.0 se entenderá que la pelota ha perdido toda la energía. • **Índice (opcional):** Tras cada rebote, variable que permite indicarle al programa que "salte" al siguiente elemento de la lista AlturasMaximas, haciéndole saber cuál debe ser la altura máxima del próximo salto.

PISTAS

- No permitas al jugador que lance la pelota si el coeficiente de restitución es menor a 0 o mayor a 1.
- Introduce como elemento 1 de la lista AlturasMaximas la altura máxima del primer salto (posición Y de la pelota * coeficiente de restitución). Aplica este mismo patrón para generar los siguientes elementos de la lista. Ten en cuenta que la altura máxima de la pelota del último salto no será la posición en Y inicial de la pelota, sino la última altura máxima registrada en la lista.

• Cuando estés registrando todos los saltos máximos, pídele al programa que cambie al siguiente elemento con la ayuda de la variable Índice. Cuando esta sea 1, hará referencia al elemento 1 de la lista, cuando sea 2, hará referencia al elemento 2… • Para hacer los rebotes de hacia arriba y hacia abajo juega con el bloque "repetir hasta que…" En el caso de estar arriba, tendrá que bajar hasta abajo continuamente hasta que el valor de Y sea inferior a 0 (mejor usar inferior y no igual, pues es más difícil para el programa captar un fotograma a captar un rango). Aplicar el mismo procedimiento para cuando suba la pelota, donde tendrá que estar continuamente subiendo hasta que la posición en Y sea mayor al elemento que corresponda de la lista de AlturasMáximas. • Para simular la Gravedad, emplea la variable Gravedad. Esta variable empezará en 0 cuando esté la pelota en el punto más alto e irá aumentando continuamente -1 (porque se aplica hacia abajo) hasta llegar a tocar el suelo.
ADICIONAL
• Permite que mientras está la pelota rebotando el jugador mueva la pelota a los laterales. • Introduce una mano arriba, simulando que el jugador mantiene la energía del último rebote hasta que se deja de tocar la pelota.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/xX2mVHMMTz4	https://bit.ly/3RiyvC5

 # LISTA DE LA COMPRA

OBJETIVO
El usuario haciendo clic sobre distintos alimentos irá introduciéndoles a la lista de la compra. Ruby, el personaje de la animación, como máximo podrá llevar un total de 15 alimentos. Al hacer clic sobre Ruby, nos dirá cuántos alimentos hemos cogido en total, y cuántos hemos cogido de cada tipo (cuántos plátanos, cuántas frases, cuántas tartas…). Un botón de reiniciar formateará el listado de la compra.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Distintos alimentos:** Se obtienen desde el repositorio de Scratch. Cada alimento será un objeto. No es necesario que cada objeto tenga más de 1 disfraz. • **Ruby:** Se obtiene desde el repositorio de Scratch. Al hacer clic sobre él nos dirá un mensaje comentando cuántos alimentos hemos cogido, además de decirnos la cantidad que hemos cogido de cada alimento. En caso de haber llegado a más de 15 elementos en el listado, lanzará un mensaje diciendo que no puede llevar más alimentos. • **Button1:** Se obtiene desde el repositorio de Scratch. Solo tendrá un único disfraz. Al hacer clic sobre él se vaciará la lista de la compra.
ESCENARIOS
(No son necesarios)
PISTAS
• Utiliza para cada alimento una condición con su contra-condición a través del bloque "Si… Si no". En ella, antes de añadir el alimento deberá verificar que no lleva más de 15 productos. • Cuando un alimento no pueda añadirse a la lista por estar llena, conecta esta función con el mensaje que dirá Ruby a través del bloque "Enviar mensaje".

SONIDOS	VARIABLES
(No son necesarios)	• **Alimentos:** Lista que albergará todos los alimentos que Ruby lleva en la lista de la compra. • **Alimentos (variables):** se creará tantas variables como alimentos se introduzcan (p. ej., tarta, leche, plátanos…). Estas variables contabilizarán la cantidad de cada alimento cuando se añada a la lista de la compra.
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/8huMSO2_Gmo	https://bit.ly/3X3KbvA

 # CIRCUITO DE CARRERAS

OBJETIVO	
En la pantalla aparecerá un circuito de carreras con curvas. El piloto será una flecha y habrá un silbato que funcionará de sensor si nos salimos del circuito o no. Además, cada vez que pasemos por meta nos sumará la cantidad de vueltas que hemos dado.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Arrow1:** Se obtiene desde el repositorio de Scratch. Solamente se necesita un único disfraz. Hará la función de piloto y nos moveremos por el circuito con las fechas del teclado. • **Referee:** Se obtiene desde el repositorio de Scratch. Se necesitan dos disfraces: uno con las manos abajo y otro enseñando la tarjeta amarilla. Cuando el piloto se salga de la pista enseñará la tarjeta amarilla. En cambio, mientras siga por el circuito estará con las manos abajo.	
ESCENARIOS	
• **Circuito:** Se puede dibujar u obtener desde internet. Es recomendable que disponga de curvas y que se diferencia a través de colores qué pertenece al circuito y qué no. También debe tener claro, a través de otro color, dónde está la salida de meta.	
SONIDOS	**VARIABLES**
(No son necesarios)	• Vueltas: Variable que va contabilizando la cantidad de veces que ha pasado el jugador por la casilla de vueltas.
PISTAS	
• Utiliza un sensor de color “tocando el color” para determinar si el jugador sigue en el circuito o se ha salido.	
ADICIONAL	
• Haz que cuando el jugador se salga del circuito, su velocidad se vea ralentizada. • Haz que cuando el jugador se salga del circuito más de una cantidad de veces determinadas, vuelva a la casilla de salida sin contabilizarle la vuelta (cuando haya dado una vuelta con pocas salidas de pista, su variable de fallos se reiniciará para cada vuelta).	

DEMOSTRACIÓN	*PROPUESTA DE SOLUCIÓN*
https://youtu.be/BByuUArIXDk	https://bit.ly/42XwJwW

 # CAZANDO AL GATO

OBJETIVO	
En la pantalla aparecerá un gato y una bomba. La bomba se moverá a una posición aleatoria cada 5 segundos. El objetivo será, con la cámara web activada, tocar al gato sin tocar la bomba. Si tocamos el gato obtendremos puntos y se moverá a una posición aleatoria. Si tocamos la bomba perderemos puntos y se moverá a una posición aleatoria.	
ELEMENTOS RECOMENDADOS DEL RETO	
OBJETOS	
• **Cat:** Se obtiene desde el repositorio de Scratch. Contiene un único disfraz. Si lo tocamos con nuestras manos se moverá a una posición aleatoria y obtendremos puntos. • **Bomb:** Se puede dibujar u obtener desde internet. Contiene un único disfraz. Si lo tocamos con nuestras manos, perderemos puntos y se moverá a una posición aleatoria.	
ESCENARIOS	
(No son necesarios)	
PISTAS	
• Utiliza la extensión “Sensor de vídeo” para que el fondo del escenario sea la imagen de la webcam. Esto permitirá que el programa detecte cuando la persona toca o no la bomba y/o el gato. • Utiliza el bloque “Movimiento de vídeo en objeto” como sensor del movimiento de la persona sobre el objeto (gato o bomba).	
ADICIONAL	
• Añade distintos objetos adicionales y que cada uno otorgue distinta cantidad de puntos. • Añade rayos láser que atraviesen el escenario en horizontal y/o vertical; y que segundos antes de cambiarse de posición avisen al jugador cuándo se van a mover (por ejemplo, poniéndose intermitentes).	
SONIDOS	**VARIABLES**
(No son necesarios)	• **Puntos:** Variable que gestionará la puntuación.
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
---	https://bit.ly/3QmFsBk

 # PIEDRA, PAPEL O TIJERAS

OBJETIVO

<table>
<tr><td colspan="2">El mítico juego del piedra, papel o tijeras con la condición de que el ordenador siempre debe ganar al jugador. El jugador, al hacer clic en su objeto (piedra, papel o tijeras) se generará la siguiente partida al azar y así infinitamente. Se dispondrá de un contador que empezará en 0 e irá sumando cada victoria del PC.</td></tr>
<tr><td colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</td></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• MiDecisión: Contiene 3 disfraces; uno con una piedra, otro con un papel, y otro con tijeras. Al hacer click sobre él, elegirá uno al azar entre los tres.
• DecisiónPC: Contiene los mismos tres disfraces que el objeto previo. En base a la decisión al azar del jugador, sacará el disfraz correspondiente para ganar la partida.</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>• VictoriasPC: Comienza desde 0. Sirve para contabilizar la cantidad de victorias del PC desde que comienza la partida.</td></tr>
<tr><td colspan="2">PISTAS</td></tr>
<tr><td colspan="2">• Utiliza el bloque “Enviar mensaje” para conectar tu decisión con la decisión del ordenador.</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">• Realiza el mismo procedimiento, pero teniendo en cuenta que la probabilidad de victoria del PC no sea del 100%, sino del 50%.
• En vez de que la elección del jugador sea al azar, considera que el jugador pueda elegir qué opción seleccionar.</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/y6Q62x-71UY</td><td>https://bit.ly/4i0xc5R</td></tr>
</table>

RETOS EXPERTOS

JUEGOS CLÁSICOS

 # MEMORY

<table>
<tr><td>OBJETIVO</td></tr>
<tr><td>Se presenta una serie de cartas dadas la vuelta y el jugador deberá ir buscando su pareja. Si se encuentra la pareja, estas cartas se eliminarán del juego. Si no se ha encontrado la pareja, se volverán a voltear las cartas. Solo puede haber un máximo de dos cartas a la vez dadas la vuelta. Si se opta por volver a iniciar una nueva partida, el posicionamiento de las cartas deberá ser diferente (por ende, la posición de las cartas es al azar).</td></tr>
<tr><td>ELEMENTOS RECOMENDADOS DEL RETO</td></tr>
<tr><td>OBJETOS</td></tr>
</table>

<table>
<tr><td colspan="2">

- **Cartas:** Se debe generar tantos objetos como cartas se quieran. Por ejemplo, 6 cartas deberán ser 6 objetos. Cada carta deberá tener dos disfraces: El primer disfraz será la parte trasera de la carta, y el segundo disfraz el anverso de la carta. El anverso de las distintas cartas puede llevar temáticas distintas: Unir palabras + dibujos en inglés; unir colores, etc.
- **Button 1:** Se añade desde el repositorio de Scratch. Solo debe tener 1 disfraz. Sirve para reiniciar la partida.

</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>(Recomendadas)
• Posición: Posición definitiva que va a ocupar una carta. Se les puede llamar Posición A, B, C, D, E…
• DadaBuelta?: Variable que controlará que si hay una carta dada la vuelta la siguiente carta se dé la vuelta también. En caso contrario, si ya hay dos cartas dadas la vuelta y no son idénticas, deberán girarse. Solo coge el valor 0 o 1.
• Correctas: Cantidad de parejas encontradas.
• Escogida: En caso de estar una carta girada permite verificar si la última carta que se acaba de girar es pareja o no de la ya girada.
• PosicionesRestantes: Lista que recoge todas las posiciones que pueden ocupar las cartas. Se pueden numerar como A, B, C, D, E…</td></tr>
<tr><td colspan="2">PISTAS</td></tr>
<tr><td colspan="2">

- Llama a cada posición con una letra específica. Por ejemplo A podrá corresponder a x= -100; y = 60; B…
- Emplea la lista PosicionesRestantes para indicarle al juego qué letras quedan restantes (las que falten ya habrán sido cogidas) y por ende qué posicionamientos de carta quedan disponibles.

</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">

- Ampliar el número de cartas, haciendo el juego más complejo.

</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/IGITyPxuiZY</td><td>https://bit.ly/416LBXz</td></tr>
</table>

 # PASAPALABRA MINI

OBJETIVO
Se dispone de una serie de letras (por ejemplo, de la A a la E). Al jugador se le irá haciendo preguntas en orden alfabético cuyas respuestas comiencen por la letra que corresponda. Para

cada pregunta, el jugador podrá responder bien, responder mal, o escribir “pasapalabra”. En caso de escribir “pasapalabra” no se le contabilizará como acierto ni error, y una vez llegado a la última palabra volverá a las letras que le falten. El jugador comenzará con una cantidad determinada de tiempo y en caso de no haber completado el rosco en esa cantidad de tiempo se detendrá y perderá el juego.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS

- **Letras:** Se debe añadir cada letra como objeto independiente. Si se forma un rosco de 5 letras, deberá haber 5 objetos; uno por letra. Cada letra deberá tener 3 disfraces: 1 para preguntas sin responder; 1 para preguntas respondidas correctamente; y 1 para preguntas respondidas incorrectamente.

ESCENARIOS

(No son necesarios)

SONIDOS	**VARIABLES**
(No son necesarios)	• **Tiempo:** Cronómetro regresivo que al llegar a 0 detendrá el juego. • **Correctas:** Contador de respuestas correctas. • **DetenerJuego:** Variable encargada de contabilizar cuántas respuestas se han respondido, bien o mal. Servirá para que al llegar al máximo de preguntas detenga el juego si aún queda tiempo restante.

PISTAS

- Necesitarás emplear condiciones dobles o triples para dar respuesta al reto. Ten en cuenta que dentro de una condición, puedes introducir otra condición.
- Antes de lanzar la pregunta, comprueba si la pregunta ha sido respondida o no. Para ello usa los bloques “Sí …. Si no…”, siendo la condición si el número de disfraz es el de la letra correcta o incorrecta. En caso de que sea así, podrá “mandar un mensaje” a la siguiente letra para que haga la misma verificación.

ADICIONAL

- Realizar el juego con mayor cantidad de preguntas.
- Realizar el juego sobre una temática específica (por ejemplo, animales, el sistema solar…).

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/ZRO5QKzjLuA	https://bit.ly/4b4BDdM

 # EL AHORCADO

OBJETIVO

En un primer momento, se le permitirá al usuario introducir una palabra. Posteriormente otras personas tendrán que ir adivinando letra por letra la palabra que había pensado la primera persona, introduciendo letras que consideren que puede tener la palabra a adivinar. Tras 7 u 8 intentos sin adivinar la palabra, el jugador perderá.

ELEMENTOS RECOMENDADOS DEL RETO

OBJETOS	
• **Dibujo del ahorcado:** Se debe dibujar. Deberá contener unos 7 u 8 disfraces. Por cada fallo que se vaya detectando se irá cambiando al siguiente disfraz. Cuando se llegue a completar el dibujo del muñeco decapitado, el jugador perderá. Cuando el juego se inicia se mostrará el dibujo solo con el soporte.	
ESCENARIOS	
(No son necesarios)	
SONIDOS	**VARIABLES**
(No son necesarios)	• Cantidad de letras: Variable que permite desglosar la cantidad de letras que tiene la palabra que se desea adivinar. • Contador: Permite enviar a la lista Letras la posición específica de una letra en una palabra. • Fallos: Variable contador con los fallos acumulados. Se inicia en 0. • Letras: Lista que recoge en orden las distintas letras de la palabra a adivinar. • Solución: Lista que verá el usuario con '*' o cualquier otro símbolo y que tras introducir una letra correcta se cambiará por la letra que corresponde de la palabra a adivinar.
DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/auTXVBXmTMw	https://bit.ly/4hTUt9E

APLICACIONES EDUCATIVAS

 # AGRUPAMIENTO DE GRUPOS

OBJETIVO
Esta aplicación permitirá introducir y quitar alumnos en una lista, como si de la lista de clase se tratase. Posteriormente, el usuario podrá elegir dividir aleatoriamente a la cantidad de estudiantes introducidos en grupos de dos, tres, cuatro o cinco.
ELEMENTOS RECOMENDADOS DEL RETO
OBJETOS
• **Símbolo +:** Se debe dibujar o introducir desde internet. Solamente tiene un único disfraz. Cada vez que se haga clic sobre él se le permitirá al usuario introducir un nuevo estudiante a la lista de clase. Se recomienda limitar el número de estudiantes de la lista a un tope (p. ej. 10). Si se ha capado el tope de estudiantes, avisar al usuario cuando se llegue al máximo. Tampoco se permitirá introducir estudiantes con el mismo nombre para saber cada cuál a qué grupo debe pertenecer. • **Símbolo -:** Se debe dibujar o introducir desde internet. Solamente tiene un único disfraz. Cada vez que se haga clic sobre él se le permitirá al usuario eliminar un estudiante específico de la lista de clase.

<table>
<tr><td colspan="2">

Número 2: Se puede introducir desde el repositorio de Scratch. Al hacer clic sobre él se mostrará una división aleatoria de la lista de clase en grupos de dos. Si hubiese estudiantes de sobra, se dividirán de manera proporcional a lo largo de los grupos.
Número 3: Ídem que el número 2, pero la división aleatoria de la lista de clase se hará en grupos de tres. Si hubiese estudiantes de sobra, se dividirán de manera proporcional a lo largo de los grupos.
Número 4: Ídem que el número 2 y 3.
Número 5: Ídem que los números previamente comentados.
Arrow1: Se introduce desde el repositorio de Scratch. Solamente es necesario un único disfraz. Solamente se muestra mientras se está visualizando el reparto de estudiantes. Al hacer clic sobre la flecha, permite al usuario volver al menú principal para añadir o quitar usuarios de la lista de clase y esconder el reparto de estudiantes.

</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>

Lista de clase: Lista que permite al usuario visualizar los estudiantes que figuran en su lista de clase.
Grupo 1: Lista con el alumnado que forma el grupo 1:
Grupo 2: Ídem que previamente.
Grupo 3: Ídem que previamente.
Grupo 4: Ídem que previamente.
Grupo 5: Ídem que previamente.
CantidadGrupos: Variable que calcula la cantidad de grupos que serán necesarios en función de la cantidad de alumnos en el listado de clase.
Resto: Variable que calcula los estudiantes que sobran en un reparto equitativo. Por ejemplo, si hay 13 estudiantes y hay que dividirlos en grupos de 4, se generarán 3 grupos y 1 estudiante estará en el resto. Este resto se irá dividiendo de manera equilibrada entre los grupos existentes.
EstudianteQueSeMueve: Variable que indica la posición del alumnado que se debe mover a un determinado grupo.

</td></tr>
<tr><td colspan="2">ADICIONAL</td></tr>
<tr><td colspan="2">

Permite que la cantidad de estudiantes que se pueda introducir sea el equiparable a una clase más normal, de unos 25-30 estudiantes. Esto supondría que se deberían generar unas 15 listas, en caso de que el usuario quiera hacer parejas.

</td></tr>
<tr><td>DEMOSTRACIÓN</td><td>PROPUESTA DE SOLUCIÓN</td></tr>
<tr><td>https://youtu.be/jGb6PlmyKis</td><td>https://bit.ly/3EJXUl6</td></tr>
</table>

OTROS

#ENCRIPTADOR/DESENCRIPTADOR

<table>
<tr><th colspan="2">OBJETIVO</th></tr>
<tr><td colspan="2">El objetivo del juego es encriptar o desencriptar mensajes. Para ello, habrá dos botones en el juego, cada uno para una funcionalidad distinta. Cada botón permitirá encriptar un mensaje o descriptarlo en función de dónde presione.</td></tr>
<tr><th colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</th></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• Botón de Encriptar: Se puede dibujar desde Scratch. Al hacer clic sobre él, preguntará al usuario para introducir una palabra o frase que considere. Posteriormente, se le pedirá que indique cuántos desplazamientos quiere darle al código (esta función precisamente hace que se encripte). Cuando el desplazamiento es 0 entonces A = 1; B = 2, C = 3, etc. Si el desplazamiento es diferente, el valor será el de 0 más la cantidad de desplazamientos. Por ejemplo, si desplazamiento = 3; entonces A = 4, B = 5, C = 6, etc. Tras conocer el texto y el desplazamiento el juego devolverá al jugador un mensaje con el código encriptado.
• Botón de desencriptar: Se puede dibujar desde Scratch. Al hacer clic sobre él, se le preguntará al jugador que vaya introduciendo dígito a dígito hasta que escriba No. Cuando la respuesta sea No, el juego entenderá que ya ha finalizado con la introducción de dígitos. Posteriormente, se le preguntará cuántos desplazamientos tiene el código a desencriptar. El procedimiento será a la inversa que el de botón de encriptar siempre teniendo presente que si el desplazamiento es 0, entonces A = 1; B = 2, C = 3…</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>• Desplazamiento: Variable que registra cuántos desplazamientos tiene el mensaje a encriptar o desencriptar.
• Índice: Variable que sirve para detectar el posicionamiento uno a uno de la letra que se está encriptando/desencriptando.
• Mensaje original: Texto introducido por el usuario.
• Mensaje cifrado: Texto modificado tras el proceso de encriptación.
• Código cifrado: Lista que se encarga de recoger todas las equivalencias entre números y letras.
• Desencriptador: Lista que recoge uno a uno todos los dígitos que se deben desencriptar.</td></tr>
<tr><th>DEMOSTRACIÓN</th><th>PROPUESTA DE SOLUCIÓN</th></tr>
<tr><td>https://youtu.be/98GfG1ONvs4</td><td>https://bit.ly/41QXnXG</td></tr>
</table>

 # DETECTOR DE NÚMEROS PRIMOS

<table>
<tr><th colspan="2">OBJETIVO</th></tr>
<tr><td colspan="2">Un número primo es un número que solamente puede dividirse de manera entera entre 1 y entre sí mismo. En este juego, al hacer clic en un botón donde aparecerá escrito “adivinador de números primos” se le pedirá al jugador que introduzca un número cualquiera. Tras enviar la respuesta el programa deberá detectar si el número introducido es primo o no, y si no es primo deberá decirle al jugador cuáles son todos los divisores de ese número para no ser un número primo.</td></tr>
<tr><th colspan="2">ELEMENTOS RECOMENDADOS DEL RETO</th></tr>
<tr><td colspan="2">OBJETOS</td></tr>
<tr><td colspan="2">• Botón: Se debe dibujar desde Scratch. Se trata de un botón que tiene el texto “Adivinador de números primos”. Al hacer clic en él se le lanzará la pregunta al jugador de qué número quiere analizar. De igual modo, también dará la respuesta al jugador una vez introducido el número a evaluar.</td></tr>
<tr><td colspan="2">ESCENARIOS</td></tr>
<tr><td colspan="2">(No son necesarios)</td></tr>
<tr><td>SONIDOS</td><td>VARIABLES</td></tr>
<tr><td>(No son necesarios)</td><td>• Cociente: Variable que calculará el cociente de la división.
• Divisor: Variable que calculará el divisor de la división. Comenzará el 1 e irá subiendo 1 a 1 hasta llevar al valor máximo que será el valor introducido por el jugador.
• Resto: Variable que calculará el resto de la división.
• ListaDivisores: Aquellas divisiones cuyo resto sea 0 servirán para añadir el divisor a la lista de “posibles divisores”.
• PosicionLista: Variable que se encarga de ir analizando la lista de posibles divisores, sumándole 1 en cada iteración.
• PosiblesDivisores: Lista que se encarga de ir guardando los divisores cuya división ha arrojado un resto igual a 0.</td></tr>
<tr><th colspan="2">PISTAS</th></tr>
<tr><td colspan="2">• Ten presente que un número primo es aquel, que solo obtiene un resto igual a 0 en los casos de dividirlo entre 1 y entre sí mismo.
• Programa un código que realice todas las divisiones posibles con números enteros. Por ejemplo, si el jugador introduce el número 10, empieza 10:1, 10:2, 10:3… Para ello, deberás tener distintas variables, como el dividendo (que vendrá dado por la respuesta del jugador), el divisor (que irá desde el 1 hasta el número del jugador como tope), el cociente, y el resto (que servirá para determinar si es un divisor exacto o no).
• Cuando la variable Resto sea igual a 0 se añadirá ese divisor a la lista de PosiblesDivisores.</td></tr>
</table>

• Tras evaluar todos los números, si la longitud de la lista PosiblesDivisores solo incluye el 1 y la respuesta del jugador, será un número primo. Por el contrario, si incluye mayor cantidad de divisores entonces no será un número primo.

DEMOSTRACIÓN	**PROPUESTA DE SOLUCIÓN**
https://youtu.be/kGH304pI0QE	https://bit.ly/4c3UeHe