Congélate

Grado 6° Guía 4

TIC



Estudiantes

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

Congélate

Grado 6° Guía 4

TIC

Estudiantes

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Yeimi Carina Murcia Yela
Ministra TIC

Giovanny Andrés López Cabezas
Viceministro de Transformación Digital

Gloria Patricia Perdomo Rangel
Viceministra de Conectividad

William Alexander Sánchez Martínez
Director (e) de Apropiación de TIC

Lorena Paola Fortich Tulena
Jefe de la Oficina Asesora de Prensa

Equipo Comunicaciones DATIC
Manuel Alejandro Guerrero Aponte
Carlos Andrés Méndez Silva

Equipo Técnico
Cristhiam Fernando Jácome Jiménez
Miller Jimmy Alarcón
Ricardo Cañón Moreno

BRITISH COUNCIL COLOMBIA

Felipe Villar Stein
Director de país

Laura Barragán Montaña
Directora de programas de Educación, Inglés y Artes

Marianella Ortiz Montes
Jefe de Colegios

Óscar Daniel Barrios Díaz
Jefe de Implementación Colombia Programa

Equipo operativo
Juanita Camila Ruiz Díaz
Jesica Lubrano Lavadera
Dayra Maritza Paz Calderón
Saúl Fabián Torres
César Augusto Herrera Lozano
Myriam Nathalia Salinas Tarazona
Paula Álvarez Peña
Elizabeth Marín Giraldo

Equipo técnico
Ana Lorena Molina Castro
Alejandro Espinal Duque
Vanesa Abad Rendón
Raisa Marcela Ortiz Cardona
Juan Camilo Londoño Estrada

Edición y revisión técnica de versiones finales
Alejandro Espinal Duque
Ana Lorena Molina Castro
Heidy Esperanza Gordillo Bogota
Juan Camilo Londoño Estrada

Apoyo en la edición y revisión académica y técnica (versión 1)
Vanesa Abad Rendón
Raisa Marcela Ortiz Cardona

Consultora experta
Heidy Esperanza Gordillo Bogota

Edición
Juanita Camila Ruiz Díaz
Alexandra Ruiz Correa

BRITISH COMPUTER SOCIETY (BCS) – Consultoría internacional

Julia Adamson
Directora General de Educación y Beneficio Público

Niel McLean
Jefe de Educación

Claire Williams
Coordinadora de Alianzas

ASOCIACIÓN COLOMBIANA DE FACULTADES DE INGENIERÍA - ACOFI

Edición general
Mauricio Duque Escobar

Coordinación pedagógica
Margarita Gómez Sarmiento
Mariana Arboleda Flórez
Rafael Amador Rodríguez

Coordinación de producción
Harry Luque Camargo

Asesoría estrategia equidad
Paola González Valcárcel

Asesoría primera infancia
Juana Carrizosa Umaña

Autoría
Alex González Jurado
Arlet Orozco Marbello
Harry Luque Camargo
Isabella Estrada Reyes
Lucio Chávez Mariño
Margarita Gómez Sarmiento
Mariana Arboleda Flórez
Mauricio Duque Escobar
Paola González Valcárcel
Rafael Amador Rodríguez
Rocío Cardona Gómez
Saray Piñerez Zambrano
Yimzay Molina Ramo



Diseño, diagramación, ilustración, y revisión de estilo



Esta obra se encuentra bajo una Licencia Creative Commons Atribución-No Comercial 4.0 Internacional. https://creativecommons.org/licenses/by-nc/4.0/

"La presente guía corresponde a una versión modificada del material originalmente diseñado por el equipo de autores de ACOFI, en tanto incorpora ajustes, actualizaciones y contribuciones posteriores realizadas por el equipo del British Council y el Ministerio de Tecnologías de la Información y las Comunicaciones, bajo la autorización correspondiente, para su publicación final".

ISBN: 978-628-7934-77-1

Prólogo

Ministerio de Tecnologías de la Información y las Comunicaciones

Estimadas y estimados educadores, estudiantes y comunidad educativa:

Colombia es un país de talento y de oportunidades que no siempre han estado al alcance de todas y todos. Entre ellas, la tecnología, que en las zonas más apartadas aún se percibe como un privilegio lejano.

Con la convicción de conectar territorios y cerrar brechas, emprendimos una transformación en el Ministerio TIC para acercarlo más a las regiones y a sus realidades.

Creemos en una tecnología que llegue con propósito a la vida de las personas, que permita cerrar brechas, abrir caminos y acompañar el desarrollo de las comunidades en todo el país.

Hoy, en ese camino, en el que ya hemos formado a cerca de un millón de colombianas y colombianos en habilidades digitales, presentamos las **Guías de Pensamiento Computacional de Colombia Programa**, una apuesta concreta para llevar estas oportunidades a las aulas del país.

Este material es el resultado de un trabajo colectivo en el que más de 2.000 docentes de 420 sedes educativas aportaron su experiencia y conocimiento para su construcción.

Además, contó con la asesoría de la *British Computer Society (BCS)*, entidad del Reino Unido con más de 20 años de experiencia en la incorporación del pensamiento computacional en el currículo educativo, gracias al convenio de cooperación internacional suscrito con el British Council Colombia.

Estas guías trazan una ruta pedagógica para que niñas, niños y jóvenes desarrollen nuevas formas de pensar, aprendan a resolver problemas y construyan soluciones. Son un recurso pensado para acompañar la labor de los y las docentes que enseñan pensamiento computacional.

Su uso requiere la mirada y la adaptación de quienes enseñan, teniendo en cuenta el contexto y las particularidades de cada estudiante.

Con ellas buscamos que las nuevas generaciones pasen de ser usuarias de la tecnología a protagonistas de su creación y transformación; que puedan comprenderla, dominarla y ponerla al servicio de sus propias vidas y de sus comunidades.

Queremos que, sin importar el lugar donde se estudie, en una zona rural, en la Amazonía o en una ciudad, existan las mismas oportunidades para desarrollar talento digital.

El gobierno del presidente Petro le cumplió a los colombianos y las colombianas y, por eso, buscamos que más niñas y jóvenes se reconozcan como creadoras, y que cada docente cuente con herramientas para transformar su aula.

Porque cuando conectamos territorios, abrimos oportunidades para que Colombia avance con equidad, desarrollo y justicia social.

Invitamos a estudiantes, docentes y comunidades a apropiarse de estas guías, a disfrutarlas y a llevarlas a su vida diaria. Porque cuando la educación digital llega a los territorios, no solo transforma el aprendizaje: transforma comunidades enteras y abre caminos para construir un mejor país.

Carina Murcia Yela

Carina Murcia Yela
Ministra de Tecnologías de la Información y las Comunicaciones
Gobierno de Colombia

Prólogo

British Council

Por más de 87 años, el *British Council* ha tenido presencia en el país, buscando construir lazos de confianza y cooperación entre el Reino Unido y Colombia. Uno de los mejores ejemplos de lo que podemos lograr a través de estas alianzas es *Colombia Programa*. Esta iniciativa, que desde hace tres años y de la mano con el Ministerio de Tecnologías de la Información y las Comunicaciones nos ha permitido alcanzar un objetivo ambicioso: ofrecer espacios de formación y recursos de alta calidad para que docentes y estudiantes de los colegios públicos del país desarrollen sus habilidades digitales y el pensamiento computacional con un enfoque de género.

Esta publicación marca un hito particularmente importante, pues con ella entregamos al país un material sin precedentes en América Latina y comparable con muy pocos a nivel mundial. Se trata de 79 guías pedagógicas que suman más de 4.000 páginas y contienen 30 sesiones por cada grado, desde transición hasta undécimo. Con estas, maestras y maestros podrán facilitar al menos una hora de clase semanal orientada al desarrollo del pensamiento computacional de sus estudiantes. Basta preguntarse qué otra área o asignatura en Colombia cuenta con un material semejante —en términos de amplitud, rigurosidad y acceso gratuito— para dimensionar la relevancia de este esfuerzo para el sector educativo y tecnológico del país.

Al abrir estas páginas impresas o navegar su versión digital, encontrarán una gran variedad de actividades que proponen a niños, niñas y jóvenes nuevas formas de acercarse a los temas más relevantes del presente, como el uso de la inteligencia artificial desde un enfoque ético y seguro. También verán cómo esta y otras temáticas se articulan con diversas áreas del conocimiento, promoviendo el trabajo colaborativo entre docentes y estudiantes y convirtiendo el pensamiento computacional en un puente entre las matemáticas, las ciencias, el arte y el lenguaje.

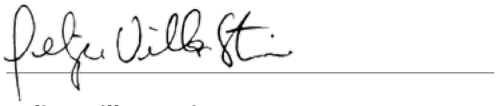
Encontrarán también en estas páginas orientaciones y sugerencias sobre cómo adaptar las actividades a diferentes contextos y necesidades. Por ejemplo, cómo desarrollar habilidades de pensamiento computacional en instituciones sin conectividad o incluso sin electricidad; cómo fomentar la equidad de género para que más niñas y jóvenes se interesen por la programación y las áreas STEM; o cómo ajustar ciertas actividades para que estudiantes en condición de discapacidad puedan participar plenamente.

Lo que quizá no sea visible a primera vista es el arduo proceso que hizo posible llegar a este punto. Un equipo de 25 personas se dedicó durante más de un año a diseñar y revisar cada una de las 79 guías. Contamos, además, con la asesoría de la *British Computer Society (BCS)*, a través de expertos que participaron y lideraron reformas curriculares en Inglaterra en el área de ciencias de la computación. Asimismo, realizamos 38 mesas de validación en cinco regiones del país, en las que 1.755 docentes revisaron y comentaron los primeros borradores. A esto se suman las innumerables horas que más de 2.100 docentes "pares expertos", pertenecientes a los 420 Nodos de Desarrollo de Pensamiento Computacional en todo el país, dedicaron en 2025 a pilotear las guías en sus aulas, asegurando así su pertinencia para los contextos educativos reales.

A cada uno de estos maestros y maestras, directivas docentes, mentores(as), tutores, expertos(as), académicos(as) y funcionarios(as) del *British Council* y de los Ministerios de las TIC y de Educación, así como a los y las estudiantes que nos acompañaron en este proceso, infinitas **gracias** por permitir que hoy Colombia cuente con esta herramienta tan poderosa.

Desde el *British Council*, sabemos que estas guías de pensamiento computacional abrirán nuevas posibilidades para que, a través del trabajo y esfuerzo de docentes en el aula, los y las estudiantes accedan a nuevas formas de pensar, de interactuar con la tecnología y de desarrollar capacidades alineadas con las demandas del mercado laboral presente y futuro. De lograrlo, conseguiremos el sueño que ha guiado esta labor y la alianza de *Colombia Programa*: que cada niña, niño y joven en Colombia pueda construir el futuro digital y material que desee.

Esperamos que las disfruten y que juntos podamos aprovechar al máximo su potencial.



Felipe Villar Stein
Director
British Council Colombia

El camino recorrido para llegar a estas guías

Creación

1

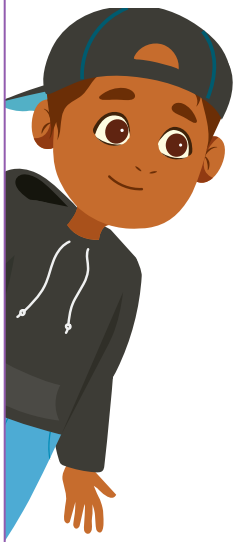
8 de noviembre de 2023
El Ministerio TIC y el *British Council* firman el convenio 1247 que da inicio a Colombia Programa.

Diciembre de 2023
La Asociación Colombiana de Facultades de Ingeniería (ACOFI) empieza a trabajar en la matriz de alcance y secuencia.

2

Febrero de 2024
Se aprueba la matriz de alcance y secuencia de las guías creadas por ACOFI, que incorpora retroalimentación de la *British Computer Society*. Con esta aprobación se inicia la creación de las guías de pensamiento computacional de Colombia Programa.

Validación y pilotaje



3

Mayo a octubre 2024
1.755 docentes en 38 mesas de trabajo de las 5 regiones del país, validan la versión inicial de las guías y proponen ajustes para hacerlas más adaptables e inclusivas.

Mayo a diciembre 2024
Se incorporan a las guías los ajustes provenientes de las mesas de validación.

4

Agosto 2024 a mayo 2025
Diagramación e impresión de las guías para entrega a los nodos de pensamiento computacional.

Marzo a septiembre 2025
2.100 docentes de los 420 nodos prueban las guías en sus aulas. Se recopilan sugerencias de ajustes a partir de este ejercicio.

Octubre a noviembre 2025
Análisis de resultados y recomendaciones para ajustes de la versión final de las guías.

Ajustes finales



5

Diciembre 2025 a abril 2026
Implementación de ajustes finales.

Enero a junio 2026
Diagramación de la versión final de las guías de pensamiento computacional de Colombia Programa.



Grado 6° Guía 4

TIC

Identificación de los aprendizajes

- Algoritmos, patrones, abstracción y descomposición
- Lógica, programación y depuración
- Prácticas de datos

Aprendizajes de la guía

Con las actividades de esta guía se espera que puedas avanzar en:

- Usar condicionales y bucles para controlar la ejecución de instrucciones.
- Utilizar condicionales para evaluar variables.
- Valorar proposiciones lógicas con al menos un conector.
- Utilizar funciones con parámetros.

Resumen de la guía

Esta guía propone 5 sesiones de trabajo que permiten aprender sobre variables booleanas, proposiciones lógicas, lógica booleana y uso de condicionales en un programa.

Se alterna entre actividades desconectadas y conectadas usando *MakeCode* y *Scratch*.

Resumen de las sesiones

Sesión 1

Se desarrolla una actividad para aprender a utilizar proposiciones sencillas de lógica booleana.

Sesión 2

Se usa un juego para practicar la utilización de proposiciones lógicas y condicionales.

Sesión 3

Se repasa el uso de *Scratch*, cambiando el comportamiento de los personajes, que se mueven aleatoriamente cuando pasan por ciertas zonas del escenario.

Si se requiere

- En la Guía 1 del grado 4° se introduce el uso del lenguaje de programación en bloques llamado *Scratch*.
- En la Guía 2 de grado 6° se trabaja una simulación de un virus cuyo código puede ser de utilidad.

Sesión 4

Se comienza a resolver el reto de la guía programando una versión sencilla del juego de la lleva.

Sesión 5

Se resuelve el reto planteado para esta guía usando los conocimientos adquiridos en las sesiones anteriores.

Conexión con otras áreas

A continuación, se presenta la conexión con otras áreas:

Matemáticas

- La utilización de proposiciones lógicas, como la lógica booleana, tiene relación con la lógica matemática.

Lenguaje y Ciencias Sociales

- La verificación de afirmaciones lógicas se usa en otras áreas como la filosofía y la argumentación.



Grado 6° Guía 4

TIC

Sesión 1

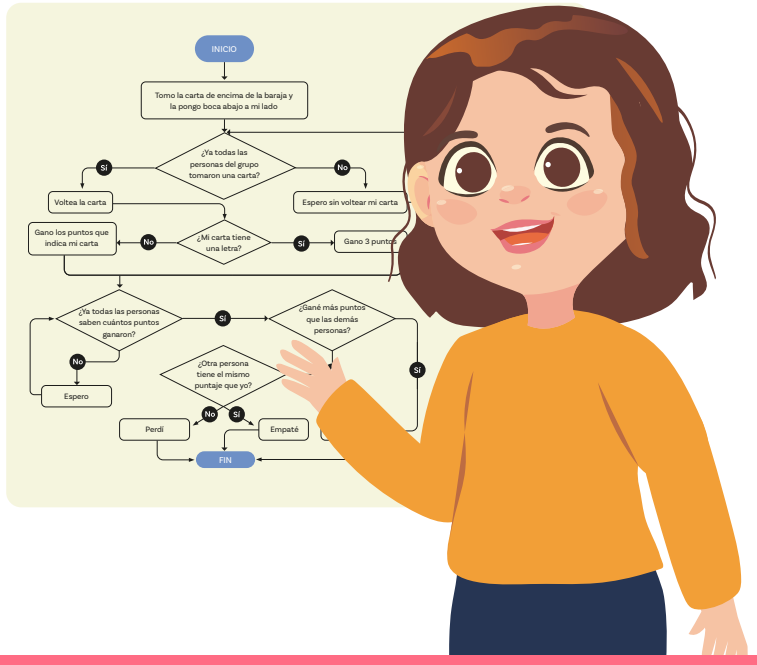
Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Proponer ejemplos de expresiones condicionales.
- Interpretar y evaluar condicionales en un algoritmo.

Duración sugerida

15% 70% 15%



Material para la sesión

Material por grupo:

- Una copia del Anexo 1.1 por estudiante.
- Una baraja de cartas de póker.
- Una copia del Anexo 1.2.
- Una hoja de papel bond o un octavo de cartulina.

Anexo

Anexo 1.1



En este reto deberás programar una simulación en *Scratch* con el tradicional juego de la lleva congelada. Las niñas y niños serán representados por objetos sencillos, por ejemplo, círculos.

Los requerimientos que debe cumplir el programa de simulación del juego son los siguientes:

1. Debe haber 4 objetos en el escenario.
2. Cada objeto debe tener un solo color.
3. 3 objetos comienzan del mismo color y 1 de un color diferente. Este último representará a la lleva.
4. Los objetos se deben mover aleatoriamente y rebotar si tocan los bordes de la pantalla.
5. La lleva debe moverse más rápido que los demás objetos.
6. El primer cuadrante del escenario debe ser de otro color y corresponde a una zona segura o "tacho" en la que la lleva no puede congelar.
7. Cuando la lleva toca a otro objeto que no esté en el "tacho", este se congela y cambia de color.
8. Cuando un objeto en movimiento, diferente de la lleva, toca a un objeto congelado, este se descongela.

Cuando tengan listo su programa, conviértanlo en un juego de computador en parejas. Con ese fin, cada jugador(a) debe hacer clic en iniciar y esperar hasta que la lleva congele los demás objetos en el escenario. Ganará la persona que logre que se congelen todos los objetos en el menor tiempo.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Empieza por leer el reto que te proponemos en esta guía, el cual se encuentra en el *Anexo 1.1.* Recuerda que debes resolverlo hasta la sesión 5. En las sesiones anteriores aprenderás lo que se necesita.

En guías pasadas ya has trabajado creando programas de computador en lenguajes como *Scratch* y *MakeCode*.

Como sabes, hay acciones que solo se realizan si se dan ciertas condiciones específicas en un programa. Por ejemplo, si se quiere emitir una alerta cuando la temperatura está muy alta, seguramente se tendrá que hacer algo como lo siguiente:

Si temperatura > 30 hacer sonar la alarma.

En la vida cotidiana también se pueden describir acciones usando condicionales. Los siguientes son algunos ejemplos:

Si es sábado, entonces no debo ir a la escuela.

Si terminé de comer, entonces debo recoger los platos.

Si voy a dormir, entonces debo ponerme la pijama.

Si terminé de lavarme las manos, entonces debo cerrar bien la llave del agua.

Si ya comí y no me he lavado los dientes, entonces debo lavármelos.

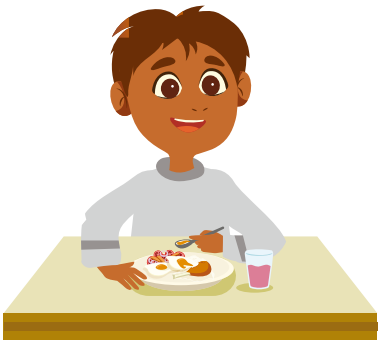
Anexo

Anexo 1.2











Muchas de las acciones que se realizan están condicionadas a que sucedan o hayan sucedido ciertas cosas.

Estas frases se escriben a menudo en el formato antes indicado:

Si (condición o condiciones), entonces (Acción)

En algunos casos resulta más apropiada una estructura más complicada, como:

Si (condición o condiciones), entonces (Acción1), si no, (Acción2)

Ahora piensa en acciones de la vida diaria que se puedan expresar en este tipo de estructura.

Observa este otro ejemplo más:

Si (es domingo y no está lloviendo), entonces (salgo a montar bicicleta), si no, (me quedo en casa).

Es tu turno, piensa en otros ejemplos y luego compártelos con la clase. En nuestra vida cotidiana con frecuencia tomamos decisiones que se podrían formalizar en frases como estas.

¿? *¿Encontraron muchos ejemplos?*

En la siguiente sección participarás en un juego de cartas con el que podrán practicar con este tipo de condicionales.

Manos a la obra

Desconectadas



Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos de 4 a 6 personas, siguiendo las indicaciones de tu docente.

En su grupo, observen el algoritmo dispuesto en forma de diagrama de flujo que aparece en el *Anexo 1.2*.

¿? *¿Para qué creen que sirve?*

Prepárense para explicarlo usando sus propias palabras. Su docente hará algunas preguntas para verificar que todas las personas del salón comprendan las reglas que allí se describen.

Cuando haya claridad sobre la forma en que se sigue el algoritmo para jugar con las cartas, empiecen a jugar siguiendo las instrucciones que ahí se describen.

Este juego les ayudará a avanzar en la habilidad de valorar expresiones lógicas.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes proponer ejemplos de expresiones condicionales?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2. ¿Puedes interpretar y evaluar condicionales en un algoritmo?

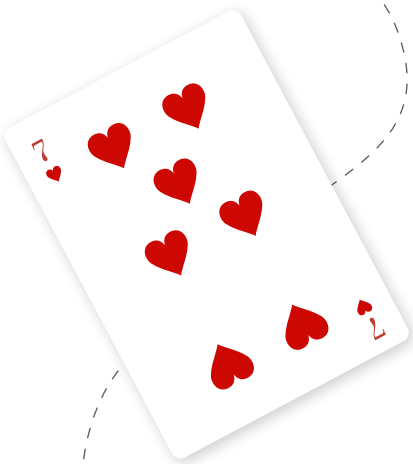
- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas. Luego, discute con tus compañeras y compañeros de grupo cómo se utilizaron los condicionales durante el juego de cartas descrito en el diagrama de flujo inicial y en el que ustedes crearon. Si todavía te quedan dudas, consúltale a tu docente.**

Ahora te proponemos un reto con tu grupo usando una rutina llamada pensar, presentar e integrar (P-P-I):

- ◯ Primero, respondemos individualmente.
- ◯ Luego, cada persona en el grupo y en su turno le presenta al resto del equipo sus respuestas.
- ◯ Finalmente, el grupo integra una respuesta unificada.







Las preguntas que te proponemos:

¿?

Si regresamos al reto, ¿en qué parte te servirán los condicionales?

Si piensas en una receta de cocina, ¿Se usan condicionales? ¿Cómo?

Aprovecha este espacio final para hacerte un esquema en el que resumas algo de lo que aprendiste, por ejemplo, colocando las definiciones de palabras y algún ejemplo de uso de los tipos de variables empleadas.

Actividad opcional para seguir consolidando aprendizajes

Prepárate para participar en un juego de preguntas que liderará tu docente.

Podrán hacerte preguntas como estas:

¿?

¿Cómo te pareció el juego de cartas del algoritmo?

¿Qué reglas cambiaron al crear el nuevo diagrama?

¿Tenían reglas parecidas a las de los juegos propuestos por los otros grupos?

Además de las cartas, ¿se te ocurre otro juego donde también se utilicen condicionales?

¿Cómo crees que te ayuda este repaso sobre condicionales para poder ir pensando en resolver el reto de la lleva congelada?

Si puedes, enséñale a tu padre, tu madre o a las personas que te cuidan el juego del *Anexo 1.2* o alguna de las otras versiones que se crearon. Así te divertirás y podrás contarles lo aprendido en esta sesión.

Sesión 2

Aprendizajes esperados

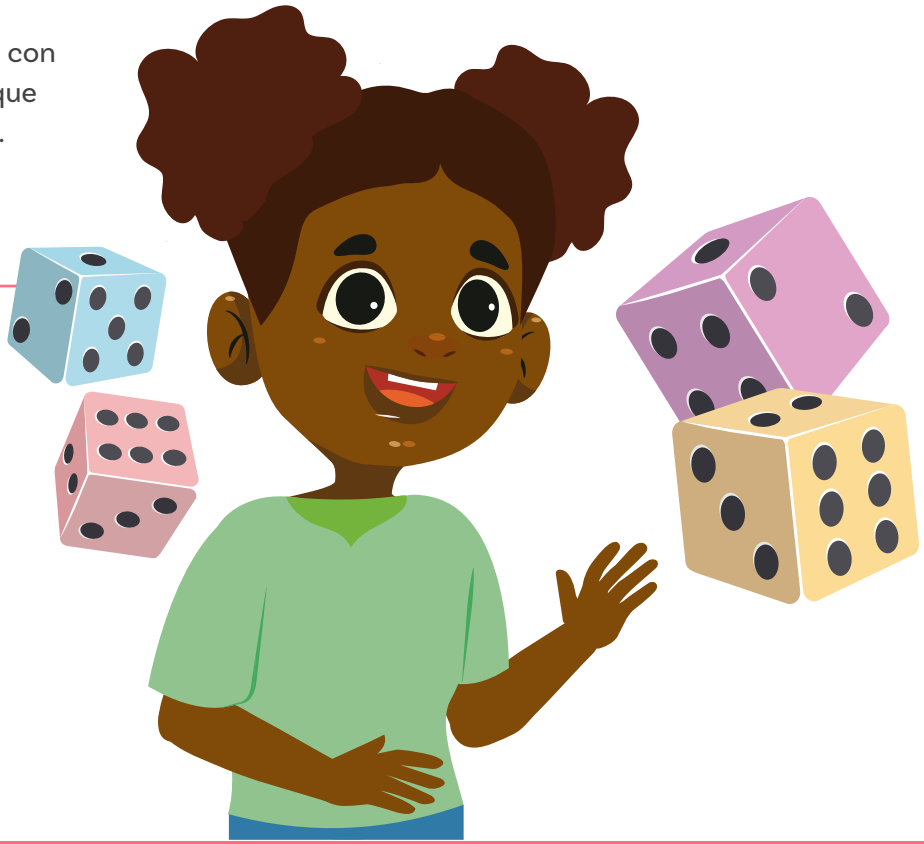
Con las actividades de esta sesión se espera que puedas avanzar en:

- Seguir algoritmos que incluyan condicionales compuestos.
- Resolver expresiones lógicas con condicionales compuestos que usen los operadores Y, O, NO.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Tarjetas recortadas de los Anexos 2.1, 2.2 y 2.3.
- ○ Tablero de juego ensamblado con las dos partes del Anexo 2.4.
- ○ 1 dado.

Material por estudiante:

- ○ Una ficha de parqués por cada miembro del grupo.

Anexos

Anexo 2.1



Anexo 2.2



Anexo 2.3



Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En la sesión anterior te divertiste jugando mientras recordabas que los condicionales permiten que se lleven a cabo algunas instrucciones dependiendo de si se cumplen o no las condiciones determinadas.

Ahora jugarás otro juego de mesa que te ayudará a practicar la toma de decisiones con **condicionales compuestos**, es decir, los que incluyen dos o más condiciones conectadas por un **operador lógico**. Además, podrás practicar la evaluación de condiciones que incluyen el operador lógico **NO**.

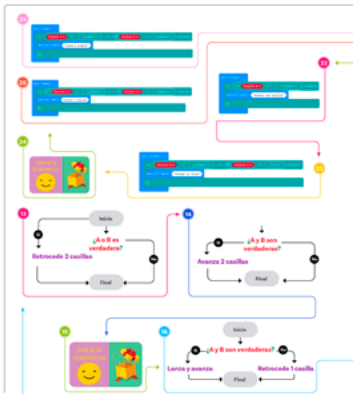
El juego de mesa requiere de los *Anexos 2.1* a *2.4*. Para jugar debes organizarte en equipos de 2 a 4 personas, según las instrucciones que dé tu docente.

Las reglas del juego son las siguientes:

- Se juega con un solo dado.
- Se deben barajar y organizar en el lugar indicado en el tablero, boca abajo, las tarjetas A, B y Sorpresa.
- Cada persona en el juego tiene una ficha que colocará en la casilla de inicio.
- Gana la persona que llegue primero a la casilla final.
- Para comenzar, todas las personas lanzan el dado y quien obtenga el número más pequeño comienza. Le sigue la persona que haya obtenido el segundo valor más pequeño y así para el resto, si juegan más de 2 personas en el equipo.
- En su turno, cada persona lanza el dado y avanza su ficha el número de casillas que corresponda al valor obtenido con el lanzamiento del dado.

Anexo

Anexo 2.4



- Al caer en una casilla, se debe leer en voz alta la condición e interpretarla utilizando las tarjetas A, B o Sorpresa y luego realizar la acción indicada según corresponda.
- Siempre se debe tomar la tarjeta que se encuentra encima del mazo. Una vez utilizada esta tarjeta se coloca debajo de todas las demás.
- Si la ficha llega a una casilla realizando una acción diferente a avanzar o retroceder el número que indica el dado, no se debe volver a evaluar la nueva condición, sino esperar su próximo turno para lanzar el dado y avanzar.

¡Hora de jugar!

Una vez hayan completado el juego, prepárense para programar en *MakeCode*.

Glosario

- **Condicional compuesto:** es cuando se usan varias condiciones juntas para tomar una decisión. Estas condiciones se conectan con operadores lógicos. Para saber si el resultado de la evaluación es verdadero o falso primero se debe evaluar cada condición por separado.
- **Operadores lógicos:** son conectores que permiten evaluar una condición, el "NO", o dos condiciones, el "Y" y el "O". Cada una de esas condiciones puede contener condiciones internas que deben evaluarse previamente.
- **Y:** operador lógico que solo da resultados verdaderos si las condiciones son verdaderas.
- **O:** operador lógico que da resultados verdaderos si por lo menos una de las dos condiciones es verdadera.
- **NO:** operador lógico que niega un valor lógico, convirtiendo uno falso en verdadero y viceversa.

Manos a la obra

Conectadas

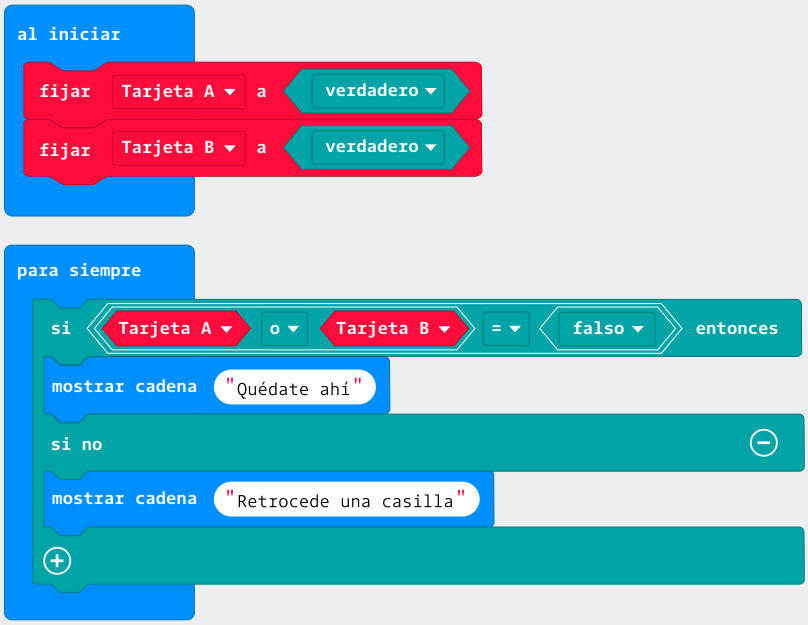
Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos, siguiendo las indicaciones de tu docente.

Su reto es programar en *MakeCode* algunos códigos como los que aparecen en el tablero de juego del *Anexo 2.4* para comprobar la evaluación de condicionales compuestos sin ayuda de dispositivos computarizados.

1. Observen el código que se presenta en la *Figura 1*.

Figura 1. Código para evaluar el resultado de la operación



Seguro notaron que el código del bloque para siempre es el mismo que está en el tablero del juego, pero ¿qué creen que hace el código que aparece en el bloque al iniciar?

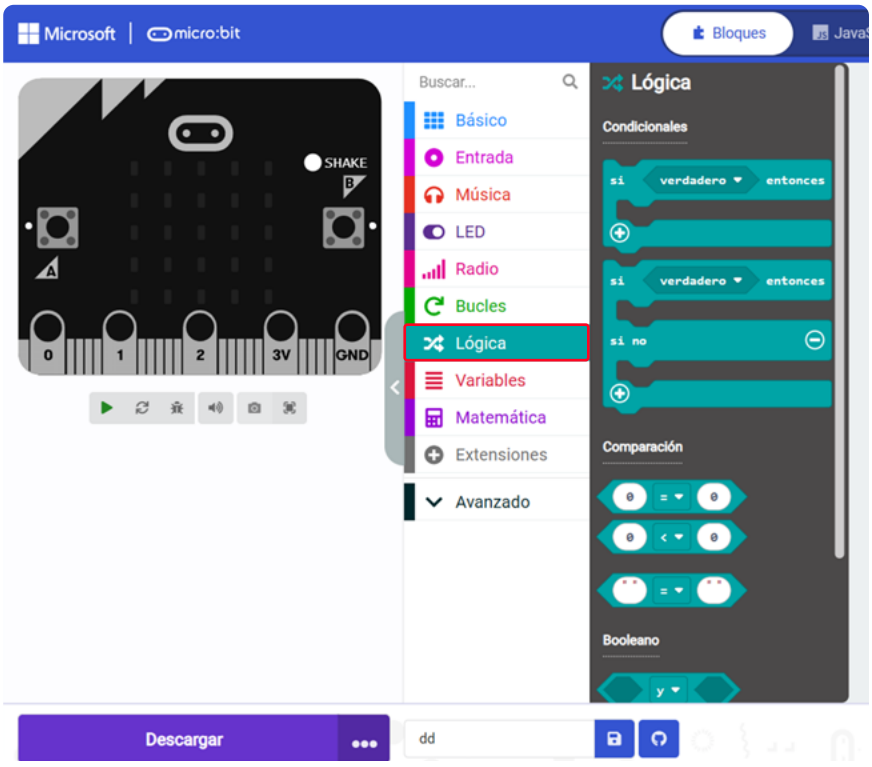
Como quizás lo notaron, los bloques Tarjeta A y Tarjeta B que aparecen en la condición son de color rojo, pues son variables creadas en *MakeCode*. Seguro también recuerdan que el bloque fijar variable a valor permite asignar a una variable un valor específico.

En otras ocasiones han visto que se asigna a las variables un valor numérico, sin embargo, como notan en la *Figura 1*, se pueden asignar **valores booleanos** a las variables.

2 Ahora sí, empiecen por replicar el código que aparece en la *Figura 1*.

Recuerden que los bloques de lógica de *MakeCode* incluyen tanto los condicionales como los operadores de comparaciones, los **valores booleanos** y los operadores lógicos, como se muestra en la *Figura 2*.

Figura 2. Bloques de lógica en *MakeCode*



Nota: recuerda que *MakeCode* cuenta con herramientas de accesibilidad como alto contraste y funciona con lectores en pantalla. No olvides pedir ayuda a tu docente si necesitas utilizar alguna de estas opciones.



3. Ejecuten el programa de la *Figura 1* en el simulador y verifiquen cuál de los dos mensajes es el que sale en pantalla cuando los valores de las dos variables (Tarjeta A y Tarjeta B) se **inicializan** como verdaderos.

4. Modifiquen el valor inicial de las variables, ver *Figura 3*, y ejecuten nuevamente el programa. Observen cuál de los dos mensajes aparece en la pantalla del simulador cuando:

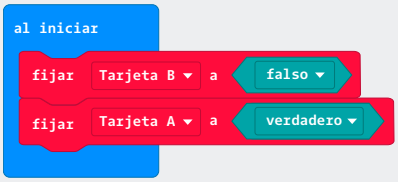
   a. solo la variable Tarjeta A es verdadera.

   b. solo la variable Tarjeta B es verdadera.

   c. ambas variables son falsas.

Escriban sus conclusiones a partir de lo observado.

Figura 3. Cambio en la inicialización de las variables



5. La *Figura 4* muestra una herramienta que permite encontrar el resultado al evaluar expresiones que hacen uso de operadores lógicos. Se le conoce como **Tabla de verdad**. Complétenla a partir de lo aprendido al ejecutar el programa de la *Figura 1*.

Figura 4. Tabla de verdad del operador lógico O

Tarjeta A	Tarjeta B	Tarjeta A o Tarjeta B
verdadero	verdadero	Verdadero
verdadero	falso	Verdadero
falso	verdadero	
falso	falso	

6 Modifiquen el código de su programa para que ahora evalúe la expresión 

.

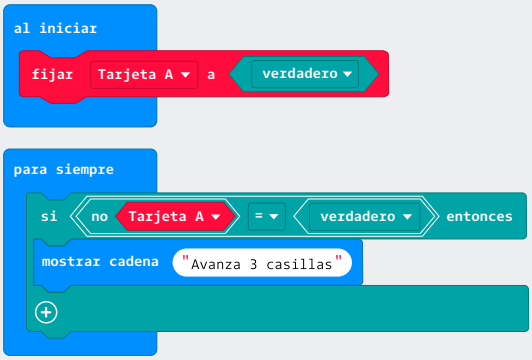
Luego, completen la tabla de verdad correspondiente.

Figura 5. Tabla de verdad del operador lógico Y

		
		
		Falso
		
		

7 Ahora creen el código de la *Figura 6*.

Figura 6. Código usando operador lógico NO



8 Cambien el valor inicial de la variable Tarjeta A y comprueben qué sucede al ejecutar el programa. Luego, completen la tabla de verdad.

9 Tomen algunas de las tarjetas A y B que se usaron en el juego de mesa y denles valores según corresponda para evaluar la expresión que aparece en el código de la *Figura 1*, tal como lo hicieron antes. Anoten su respuesta. Después, revisen sus respuestas haciendo uso de las tablas de verdad.

¿? *¿Encontraron alguna diferencia?*

Glosario

Valor booleano: valor que solo puede ser verdadero o falso. También se le conoce como valor de verdad.

Inicializar variables: asignar a un valor a las variables para que lo carguen en memoria al empezar un programa.

Tabla de verdad: herramienta usada para encontrar el resultado de evaluar condicionales compuestos. En esta tabla se incluyen todas las combinaciones posibles de valores de las condiciones y se indica si el resultado es verdadero o falso en cada caso.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1 ¿Puedes seguir algoritmos que incluyan condicionales compuestos?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2 ¿Puedes resolver expresiones lógicas con condicionales compuestos que usen los operadores Y, O, NO?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas. Luego, discute con tus compañeras y compañeros de grupo algunos de los ejemplos de condicionales que aparecen en el juego de mesa. Identifica las condiciones individuales, los operadores lógicos y los resultados que generan. Si todavía te quedan dudas, consúltale a tu docente.

Ahora te proponemos un reto con tu grupo usando una rutina llamada pensar, presentar e integrar (P-P-I):

- ◯ Primero, respondemos individualmente.
- ◯ Luego, cada persona en el grupo y en su turno le presenta al resto del equipo sus respuestas.
- ◯ Finalmente, el grupo integra una respuesta unificada.

Las preguntas que te proponemos:

¿?

Si revisamos el reto inicialmente planteado, ¿dónde se podrían requerir condicionales?

¿Puedes identificar alguna acción en la vida cotidiana que dependa de tres condiciones o más?

Aprovecha este espacio final para hacerte un esquema en el que resumas algo de lo que aprendiste, por ejemplo, colocando las definiciones de palabras y algún ejemplo de uso de los tipos de variables empleadas.

Sesión 3

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Cambiar el disfraz de los objetos en *Scratch*, en función de su posición en el escenario.
- Usar funciones cuando se requiere utilizar el mismo código más de una vez.
- Programar objetos que tienen el mismo comportamiento como clones.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Anexo 3.1.
- ○ Dispositivo con acceso a *Scratch*.

Anexo

Anexo 1.1



En este reto deberás programar una simulación en *Scratch* con el tradicional juego de la lleva congelada. Las niñas y niños serán representados por objetos sencillos, por ejemplo, círculos.

Los requerimientos que debe cumplir el programa de simulación del juego son los siguientes:

1. Debe haber 4 objetos en el escenario.
2. Cada objeto debe tener un solo color.
3. 3 objetos comienzan del mismo color y 1 de un color diferente. Este último representará a la lleva.
4. Los objetos se deben mover aleatoriamente y rebotar si tocan los bordes de la pantalla.
5. La lleva debe moverse más rápido que los demás objetos.
6. El primer cuadrante del escenario debe ser de otro color y corresponde a una zona segura o "tacho" en la que la lleva no puede congelar.
7. Cuando la lleva toca a otro objeto que no esté en el "tacho", este se congela y cambia de color.
8. Cuando un objeto en movimiento, diferente de la lleva, toca a un objeto congelado, este se descongela.

Cuando tengan listo su programa, conviértanlo en un juego de computador en parejas. Con ese fin, cada jugador(a) debe hacer clic en iniciar y esperar hasta que la lleva congele los demás objetos en el escenario. Ganará la persona que logre que se congelen todos los objetos en el menor tiempo.

Figura 1. Animales que se camuflan







Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Para desarrollar las actividades que se proponen en esta guía necesitarás hacer uso de algunas habilidades en *Scratch* que practicaste con la Guía 2. Entre ellas, crear objetos (personajes), clonar objetos, fijar una posición inicial y un movimiento aleatorios, rebotar en los bordes, cambiar de color al tocar otro personaje y el uso de funciones te serán de gran utilidad en esta y las próximas sesiones. Si necesitas reforzar alguna de estas habilidades, regresa nuevamente a esa guía y al código que creaste para simular el contagio del virus.

Ahora sí, dale una mirada al reto de esta guía, que aparece en el *Anexo 1.1*.

En cada una de las siguientes tres sesiones vas a ir adquiriendo una parte de los aprendizajes necesarios en *Scratch* para poder resolver el reto. En esta sesión aprenderás cómo cambiar el comportamiento de los objetos cuando pasen por una zona del escenario.

Manos a la obra Conectadas

Esta sección corresponde al 85% de avance de la sesión

Reúnete en grupos de 2 a 3 personas, según lo indique tu docente.

Su reto será crear un programa en *Scratch* que simule el mimetismo de animales como los camaleones, las ranas y los pulpos, ver *Figura 1*. El mimetismo, como quizás lo han aprendido, es la capacidad de cambiar el color de la piel para camuflarse con el entorno. Esta adaptación les permite a estos animales protegerse de sus depredadores.

Nota

Recuerda que *Scratch* funciona sin conectividad si está instalado en un computador e igualmente puede accederse desde celulares o tabletas con Android o IOS. Además, este programa puede trabajarse con lectores en pantalla como *NVDA* y *JAWS*. Si necesitas hacer uso de esta característica, pide apoyo a tu docente.

Enlace

Recuerda que puedes acceder a *Scratch* en:

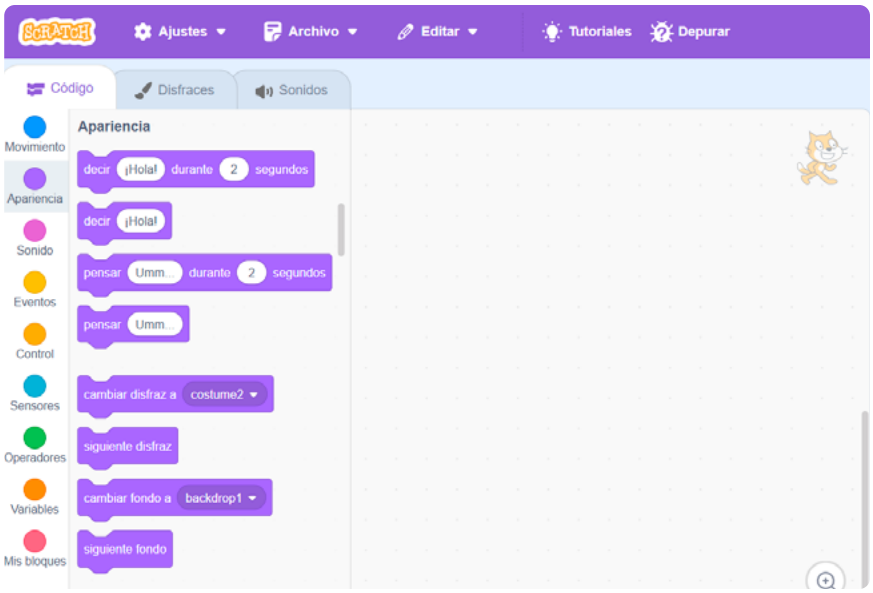
Por tanto, su programa requerirá que los objetos o personajes se camuflen, cambiando de color al pasar por diferentes zonas del escenario.

Ese programa debe cumplir con los siguientes requerimientos:

1. Debe haber 24 objetos en el escenario.
2. Cada objeto solo puede tener un color a la vez.
3. Todos los objetos deben empezar del mismo color.
4. Los objetos se mueven aleatoriamente y rebotan en los bordes.
5. El escenario debe ser blanco, pero los cuadrantes I y III deben ser de colores diferentes. Por tanto, el escenario parecerá un plano cartesiano.
6. Cuando los objetos pasan por esos cuadrantes cambian de color a uno similar al del cuadrante.

Los requerimientos 5 y 6 son novedosos. Aprenderán primero cómo responder a estos.

Figura 2. Menú de herramientas de *Scratch*



En *Scratch* se pueden cambiar, modificar o incluso usar varios escenarios en un mismo programa.

Como saben, al abrir un nuevo proyecto el escenario es un fondo blanco. Si pasan el puntero sobre el cuadro de la esquina inferior derecha que dice **Escenario**, el borde se resalta a color.

1. Hagan clic para seleccionarlo. Al hacer clic se darán cuenta de que el código disponible cambió, como pueden ver en la *Figura 2*. Esto es porque el escenario también puede tener su propio código, pero con algunas diferencias con respecto a las herramientas disponibles para los objetos. Por ejemplo, un escenario no se puede mover.

2. Den clic en la pestaña fondos. Deben elegir un color y hacer un rectángulo que ocupe el primer cuadrante. Luego cambien de color y dibujen otro rectángulo que ocupe el tercer cuadrante. La *Figura 3* muestra un ejemplo de cómo debería quedar el escenario al seguir estas indicaciones.

   Si ya lo hicieron, ya aseguraron que su programa responde al quinto requerimiento del programa de simulación del mimetismo animal.

Puedes hacer un rectángulo inicial y luego cambiarle el tamaño y desplazarlo para que quede ocupando el cuadrante. Para asegurarte de que coincida bien con los ejes, puedes hacer zoom y usar la guía que aparece en el centro.

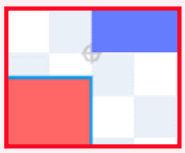
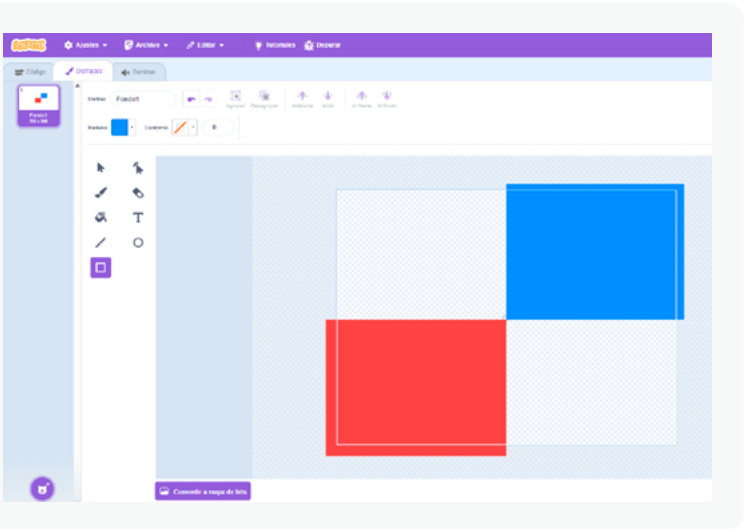




Figura 3. Ejemplo del escenario



3 Para responder al sexto requerimiento, lo primero es poder determinar cuándo un objeto se encuentra en el primer y tercer cuadrantes.

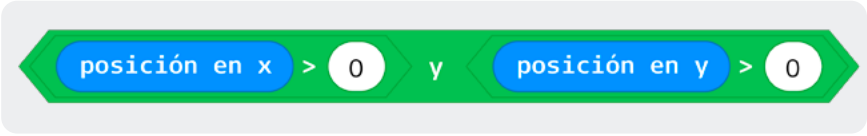
¿? *¿Se les ocurre alguna manera?*

Un punto en el primer cuadrante del plano cartesiano tiene ambas coordenadas (x, y) positivas.

Es decir que x > 0 y y > 0.

En *Scratch* esto se implementa como se muestra en la *Figura 4*, utilizando bloques del grupo **Operadores** y **Movimiento**.

Figura 4. Bloque de operadores y movimiento



Utilizando la expresión de la *Figura 4* en un condicional, pueden realizar una acción si el objeto se encuentra en el primer cuadrante. En este caso, la acción es un cambio de disfraz, que es la forma en la que se le puede cambiar el color al objeto.

El código en la *Figura 5* es un ejemplo que muestra cómo cambiarle el disfraz al personaje gato.

Si (x > 0) y (y > 0), entonces cambiar al disfraz2.

Figura 5. Código en *Scratch* para cambiar el disfraz

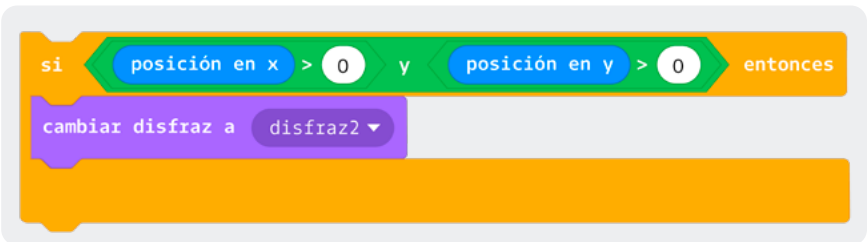


Figura 6. Ejemplo de objeto sencillo en *Scratch*

Ya pueden detectar cuándo un objeto está en el primer cuadrante y cambiarle su disfraz. Ahora deben tener en cuenta los requerimientos adicionales que tienen los objetos. En este caso no necesitan el objeto gato, así que bórrenlo y dibujen un nuevo objeto. Recuerden que el objeto solo puede tener un color. Por tanto, pueden crear algo sencillo como el objeto de ejemplo que se muestra en la *Figura 6*.

4. Nombren ese disfraz como "Inicio", ya que será el disfraz con el que comenzarán todos los objetos. Luego, dupliquen el disfraz dos veces y cambien el color.

5. Nombren los disfraces con los colores correspondientes, por ejemplo "Rojo" y "Azul".

6. Ahora analicen el código que aparece en el *Anexo 3.1*.

¿?

¿Qué creen que hace?

¿Qué ventaja tiene usar la función posición inicial?

¿Qué bloques se podrían incluir en otra función?

7. Implementen ese código cambiando los nombres de los disfraces para que coincidan con los de su programa, de ser necesario. Luego, pruébenlo.

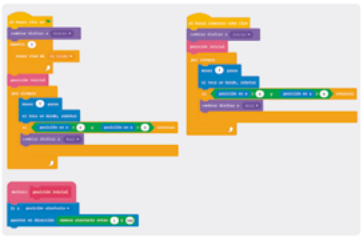
¿?

¿Qué le falta a ese programa para responder a todos los requerimientos?

8. Modifiquen el código para que cumpla con lo demás. Al hacerlo usen otra función para reemplazar las instrucciones que se repiten.

Anexo

Anexo 3.1



Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes cambiar el disfraz de los objetos en *Scratch* en función de su posición en el escenario?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes usar funciones cuando se requiere utilizar el mismo código más de una vez?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

3. ¿Puedes programar objetos que tienen el mismo comportamiento como clones?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en *Scratch*. Luego, discute con tus compañeras y compañeros de grupo lo que hace cada bloque dentro del código creado en el programa de simulación de mimetismo. Si todavía te quedan dudas, consúltale a tu docente.**

Ahora reúnete nuevamente con tu grupo para que sigan la rutina Pensar, Presentar e Integrar (P-P-I). Recuerden que esta actividad requiere que cada persona responda las preguntas planteadas de forma individual y luego comparta sus respuestas con el resto del grupo.

Posteriormente, como equipo, deben consolidar una respuesta que puedan compartir con la clase cuando su docente les pregunte.

Consideren las siguientes preguntas:

¿Cómo les permitió el programa propuesto simular el camuflaje animal?

¿Qué les llamó más la atención de la práctica realizada?

¿En qué otros casos puede servirle a un programa saber la posición de los objetos que está manipulando?

¿Cómo les sirve la actividad realizada para resolver parte del reto de la lleva congelada que se propone en el Anexo 1.3?

Aprovecha este espacio final para hacerte un esquema en el que resumas algo de lo que aprendiste, por ejemplo, colocando las definiciones de palabras y algún ejemplo de uso de los tipos de variables empleadas.



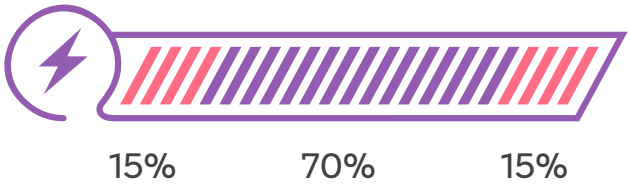
Sesión 4

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Utilizar los disfraces de los objetos como estados.
- Asociar acciones a cada estado.
- Cambiar el estado de los objetos al entrar en contacto con otro objeto.
- Usar funciones con parámetros.

Duración sugerida





Material para la sesión

Material por grupo:

- ○ Anexo 4.1.
- ○ Dispositivo con acceso a *Scratch*.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En esta sesión vas a programar una versión simplificada del juego de la lleva congelada, que es el reto central de esta guía.

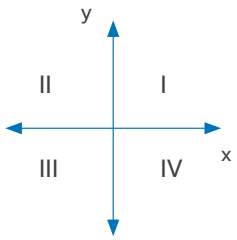
Los requerimientos de este programa son:

1. Cada objeto solo puede tener un color a la vez.
2. 5 objetos comienzan del mismo color y 1 de un color diferente (la lleva).
3. Los objetos se mueven aleatoriamente y rebotan en los bordes.
4. La lleva va más rápido que los demás.
5. Cuando la lleva toca a otro objeto, este se congela y cambia de color.

Después de leer los requerimientos, ¿conoces otros nombres para esta dinámica de juego?

Glosario

Plano cartesiano: sistema de referencia con dos líneas perpendiculares que se intersecan en el origen. Cada punto en el plan se ubica con dos números que representan la coordenadas horizontales y verticales, usualmente llamadas (x, y). El plano está dividido por los ejes en 4 cuadrantes, como se muestra a continuación:



Clonar: crear una copia idéntica de un objeto, archivo, repositorio o estructura de datos.

Manos a la obra
Conectadas

Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos de 2 a 3 personas, siguiendo las recomendaciones de tu docente.

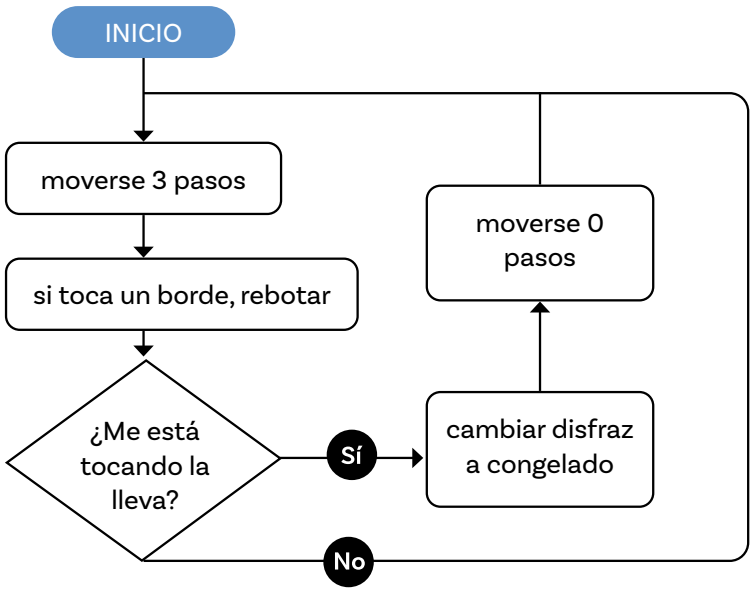
En la sesión anterior realizaron un programa de simulación que incluía requerimientos muy similares a los requerimientos 1, 2 y 3 del juego de lleva que van a programar. Así que ya saben cómo desarrollar estos tres puntos.

Ahora verán cómo responder al quinto requisito. Deben conservar la misma estructura, en la que hay un objeto que se **clona** y dos funciones. Una que indica la posición inicial y otra que determina el movimiento y cambio de disfraz.

En la *Figura 1* verán el diagrama de flujo de una primera aproximación a la función que controla el movimiento de los objetos para esta actividad.

Analicen este diagrama de flujo.

Figura 1. Diagrama de flujo: controlar el movimiento de los objetos



Recuerden que esta función le debe servir a todos los objetos, incluyendo al que tiene el rol de ser la lleva.

¿Le faltará algo a este algoritmo?

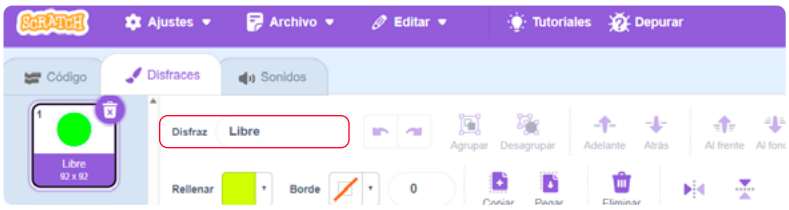
Tengan en cuenta cuándo se termina el algoritmo.

Nota

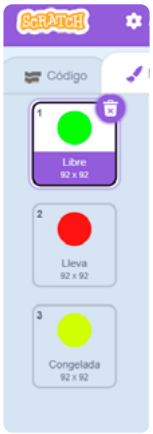
En esta actividad los objetos se comportan diferente dependiendo del rol que tengan. Algunos roles son para toda la actividad, como el de la lleva, que no cambia, y otros van cambiando pues pueden estar libres o congelados. En programación estos roles se suelen llamar estados.

En *Scratch*, una forma de implementar estados es con los disfraces.

Estos tienen 3 características que se pueden asociar a cada estado: un número, un nombre y colores. El nombre lo escribes:

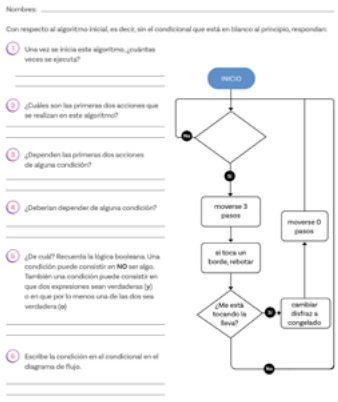


Al cambiar el orden de los disfraces, cambia el número y los colores los cambias en el lienzo. Para obtener algo así:



Anexo

Anexo 4.1



Sigan las instrucciones del *Anexo 4.1* para encontrar lo que falta.

Ahora que ya conocen el algoritmo, vean qué bloques en *Scratch* pueden ser útiles para implementarlo.

1. Lean la nota para entender el concepto de estado y ver cómo se implementa en *Scratch*.

2. Para identificar el estado de un objeto en su propio código se puede utilizar el número o el nombre. Siguiendo el ejemplo de la nota, para determinar si el estado del objeto es Libre se pueden utilizar cualquiera de las expresiones equivalentes que aparecen en la *Figura 2*.

Figura 2. Bloques de operadores en *Scratch* para identificar el estado de un objeto según su disfraz



Observen que no se puede usar la igualdad para comparar dos textos, sino que se debe preguntar si un texto está incluido en otro texto.

3. En *Scratch* solo hay una forma para determinar el estado de un bloque al que se está tocando: el color. Así que se debe utilizar el bloque sensor que aparece en la *Figura 3*.

Figura 3. Bloque de la categoría sensores **¿tocando el color __?**





Con lo anterior, ya pueden responder el quinto requisito.

4. Ahora aprenderán cómo responder al cuarto requisito. Recuerden que una única función determina el movimiento de todos los objetos, incluido la lleva. Es precisamente esa la función para la cual programaron el algoritmo anterior.

¿Qué parte de ese algoritmo determina la velocidad a la que se mueven los objetos?

¿Para modificar la velocidad necesitarían cambiar bloques completos o solo un valor dentro de estos?

Sería deseable tener la misma función, pero usando un valor diferente en el código en su interior cada una de las dos veces que se usa la función. Es decir que, cuando se llame la función para la lleva, se le pasa un valor que haga que se mueva más rápido y, cuando se llame para los otros personajes, se le pasa un valor diferente que haga que se muevan más despacio.

La función que usaron en la sesión anterior no permite ninguna entrada, pero esto es posible. En el momento de crear la función se le pueden agregar entradas, permitiendo que la función sea más versátil. Las entradas también se conocen como **parámetros**.

Al momento de crear el bloque de la función, hagan clic en **Añadir una entrada**. Al hacerlo les aparecerá un recuadro dentro del bloque. En este deben escribir el nombre que adquirirá el parámetro dentro de la función, como se muestra en la *Figura 4*.

Figura 4. Creación de la función bucle movimiento, incluyendo el parámetro Rapidez

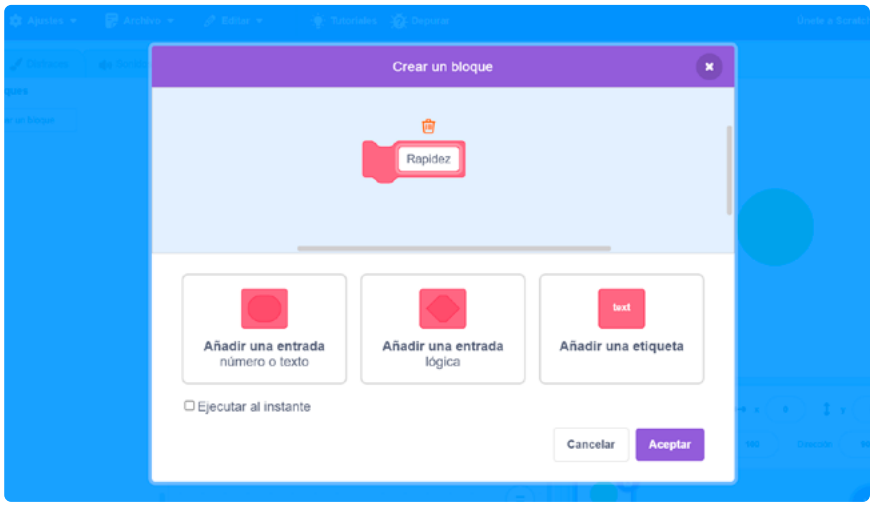


Figura 5. Función bucle movimiento

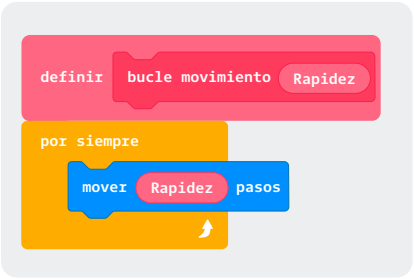
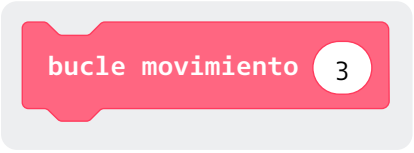


Figura 6. Bloque de llamado a la función bucle movimiento y parámetro Rapidez (3)



Una función con parámetros de todas formas correrá el mismo código, con la diferencia de que, en alguna parte, el código usará una variable que tomará el valor de la entrada. En este ejemplo, cada vez que se quiera usar la entrada dentro del bloque se usará el parámetro llamado Rapidez. Ver la *Figura 5*.

Por lo anterior, cuando se usa la función, no basta con llamarla, sino que además se le debe incluir el valor de la entrada, ver *Figura 6*.

Ya tienen todos los elementos para responder a las especificaciones del simulador de juego de lleva que les han pedido realizar en esta sesión.

Glosario

Parámetro: variable especial que se usa dentro de una función para recibir información del exterior. De esta forma al llamar una función se le pueden pasar variables diferentes en cada uso de la función.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes utilizar los disfraces de los objetos como estados?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2. ¿Puedes asociar acciones a cada estado?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

3. ¿Puedes cambiar el estado de los objetos el entrar en contacto con otro objeto?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

4. ¿Puedes usar funciones con parámetros?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en *Scratch*. Luego, revisa con tus compañeras y compañeros de grupo lo que hicieron al crear la simulación del juego de lleva, en su versión simplificada. Si todavía te quedan dudas, consúltale a tu docente.**

Ahora te proponemos un reto con tu grupo usando una rutina llamada pensar, presentar e integrar (P-P-I):

- ◯ Primero, respondemos individualmente.
- ◯ Luego, cada persona en el grupo y en su turno le presenta al resto del equipo sus respuestas.
- ◯ Finalmente, el grupo integra una respuesta unificada.

Las preguntas que te proponemos:

> *Si regresamos al reto, lo que hemos aprendido en esta sesión ¿para qué serviría? Usar funciones ¿qué ventajas trae?*

Aprovecha este espacio final para hacerte un esquema en el que resumas algo de lo que aprendiste, por ejemplo, colocando las definiciones de palabras y algún ejemplo de uso de los tipos de variables empleadas.

Actividad opcional para seguir consolidando aprendizajes

Trabaja ahora con tu grupo. Creen, al menos, 5 preguntas de repaso con al menos 3 opciones de respuesta, pero solo una respuesta correcta. Estas preguntas deben destacar lo que han aprendido o recordado con las actividades desarrolladas a lo largo de esta guía.

Marquen la respuesta correcta de cada una de sus preguntas. Cuando las tengan listas, entréguenselas a su docente. Algunas de sus propuestas podrán ser posteriormente utilizadas por su docente para hacer un juego de revisión de lo aprendido con toda la clase.

Sesión 5

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Implementar funciones para la programación de condicionales.
- Programar interacciones entre objetos considerando estado, posición y tiempo.
- Usar operadores lógicos y bloques avanzados para decisiones.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por estudiante:

- ○ Anexo 1.1.

Material por grupo:

- ○ Dispositivo con acceso a *Scratch*.

Anexo

Anexo 1.1



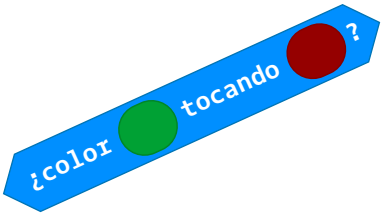
En este reto deberás programar una simulación en Scratch con el tradicional juego de la lleva congelada. Las niñas y niños serán representados por objetos sencillos, por ejemplo, círculos.

Los requerimientos que debe cumplir el programa de simulación del juego son los siguientes:

1. Debe haber 4 objetos en el escenario.
2. Cada objeto debe tener un solo color.
3. 3 objetos comienzan del mismo color y 1 de un color diferente. Este último representará a la lleva.
4. Los objetos se deben mover aleatoriamente y rebotar si tocan los bordes de la pantalla.
5. La lleva debe moverse más rápido que los demás objetos.
6. El primer cuadrante del escenario debe ser de otro color y corresponde a una zona segura o "tacho" en la que la lleva no puede congelar.
7. Cuando la lleva toca a otro objeto que no esté en el "tacho", este se congela y cambia de color.
8. Cuando un objeto en movimiento, diferente de la lleva, toca a un objeto congelado, este se descongela.

Cuando tengan listo su programa, conviértanlo en un juego de computador en parejas. Con ese fin, cada jugador(a) debe hacer clic en iniciar y esperar hasta que la lleva congele los demás objetos en el escenario. Ganará la persona que logre que se congelen todos los objetos en el menor tiempo.







Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Ya has aprendido casi todo lo necesario para resolver el reto propuesto al comienzo de la guía. Eso incluye, entre otras:

1. Crear objetos.
2. Clonar objetos.
3. Crear funciones con parámetros para generar el movimiento de los objetos.
4. Determinar la posición de un objeto en el escenario.
5. Cambiar el disfraz de un objeto, dependiendo de su ubicación en el escenario.
6. Utilizar los disfraces como forma de determinar los estados de un objeto.
7. Utilizar parámetros para asignar diferentes valores de velocidad de movimiento a un objeto, dependiendo de su estado.

En esta sesión vas a resolver el reto de crear un programa que simule el juego de la lleva congelada, cumpliendo con todos los requerimientos que se solicitan. Toma unos minutos para leer de nuevo el reto que aparece en el *Anexo 1.1* y prestar atención a estos requerimientos.

Figura 1. Operador lógico NO en *Scratch*



Figura 2. Bloque de la categoría Sensores. ¿color ___ tocando color ___?



Manos a la obra

Conectadas

Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos de 2 a 3 personas, siguiendo las recomendaciones de tu docente.

Ingresen a *Scratch*. Antes de empezar a resolver el reto, van a revisar unos últimos detalles que necesitan conocer.

1 **Determinar que un objeto no se encuentra en una ubicación específica.**

Ya saben qué expresiones usar para determinar si un objeto se encuentra en un cuadrante. Como lo vieron en la sesión 3, necesitan evaluar, por medio de condicionales, los valores en las coordenadas x,y que correspondan a la posición del objeto.

¿?

¿Cómo harían para determinar que el objeto no se encuentra en un cuadrante?

Una forma de hacer justo esto es evaluar que el objeto no esté en el cuadrante. Para esto, se usa el operador lógico **no** que está en el grupo de bloques **Operadores** y que se muestra en la *Figura 1*. Tengan en cuenta que en el interior de este bloque deben incluir la expresión o expresiones que quieren evaluar para determinar si son falsas.

2 **Utilizar un único bloque para evaluar si un color específico toca a otro.**

Si bien ya saben cómo determinar en el código de un objeto su propio estado y el estado de otro objeto que toca, *Scratch* tiene un bloque que hace ambas cosas. Este es el bloque que se presenta en la *Figura 2* y se encuentra en el grupo de bloques **Sensores**. Como observan, usa el color para determinar el estado de ambos objetos.

Figura 3. Bloque de la categoría Sensores que activa un cronómetro en pantalla

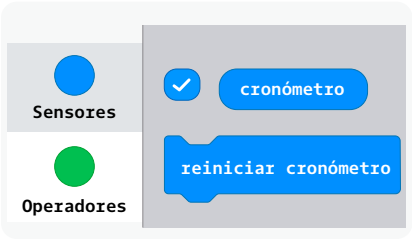
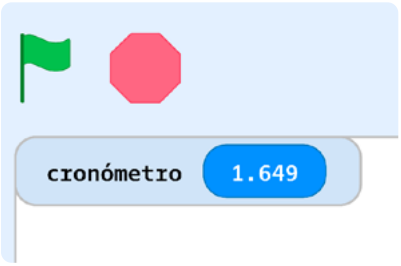


Figura 4. Cronómetro que aparece en pantalla al iniciar el programa



3 Mostrar en la pantalla el tiempo de funcionamiento del juego.

El reto de la guía requiere conocer el tiempo que ha transcurrido desde que se inició la ejecución del programa. Para esto activen el bloque cronómetro que pertenece a la categoría de bloques **Sensores**, ver *Figura 3*. Al activarlo aparecerá una ventana con un cronómetro en la esquina superior izquierda del escenario, como pueden ver en la *Figura 4*.

¡Con esto ya tienen todos los conocimientos requeridos para resolver el reto de la guía! Trabajen como equipo para ir creando el programa y dando cumplimiento a cada uno de los requerimientos.

Al terminar su código, autoevalúen su trabajo en una escala de ★ a ★★★★★, considerando los siguientes aspectos.

No.	Descriptor	Valoración en estrellas
1	Nuestra simulación del juego de la lleva congelada cumple los 8 requerimientos solicitados en el reto.	
2	Utilizamos los bloques nuevos que se presentaron en esta sesión.	
3	Todas las personas del equipo ayudamos a crear el código.	

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes implementar funciones para la programación de condicionales?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes programar interacciones entre objetos considerando estado, posición y tiempo?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

3. ¿Puedes usar operadores lógicos y bloques avanzados para decisiones?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", revisa nuevamente la valoración de estrellas que le dieron a su código. Pregúntales a tus compañeras y compañeros de grupo por lo que hacen los bloques que aparecen en su programa y que sientas que aún no comprendes totalmente. Toma notas en tu cuaderno. Si todavía te quedan dudas, consúltale a tu docente.**

Ahora te proponemos un reto con tu grupo usando una rutina llamada pensar, presentar e integrar (P-P-I):

- Primero, respondemos individualmente.
- Luego, cada persona en el grupo y en su turno le presenta al resto del equipo sus respuestas.
- Finalmente, el grupo integra una respuesta unificada.

Las preguntas que te proponemos:

Si regresamos al reto, lo que hemos aprendido en esta sesión, ¿qué fue difícil? ¿Por qué?

¿Si retomas el reto del virus de la guía 2 de grado 6°, podrías ahora incluir una condición en la que las personas enfermas se recuperan para tener un modelo más exacto y predecir lo que pasa en la propagación de un virus?



La computación y la sociedad

En esta guía resolviste un reto relacionado con un juego, pero, en el fondo, te dio los conocimientos necesarios para mejorar el programa de simulación del virus que se trabajó en la Guía 2. Podrías, por ejemplo, incluir en la simulación pacientes recuperados o en aislamiento físico por contagio.

La computación es una herramienta poderosa para analizar situaciones en la sociedad.

De forma similar a como se propaga un virus, se pueden modelar otras situaciones, como la circulación de noticias falsas.

Personas que saben de computación tienen grandes posibilidades en el trabajo, dado que estas habilidades son cada vez más importantes.

Anexo 1.1 Reto La lleva congelada

En este reto deberás programar una simulación en *Scratch* del tradicional juego de la lleva congelada. Las niñas y niños serán representados por objetos sencillos, por ejemplo, círculos .

Los requerimientos que debe cumplir el programa de simulación del juego son los siguientes:

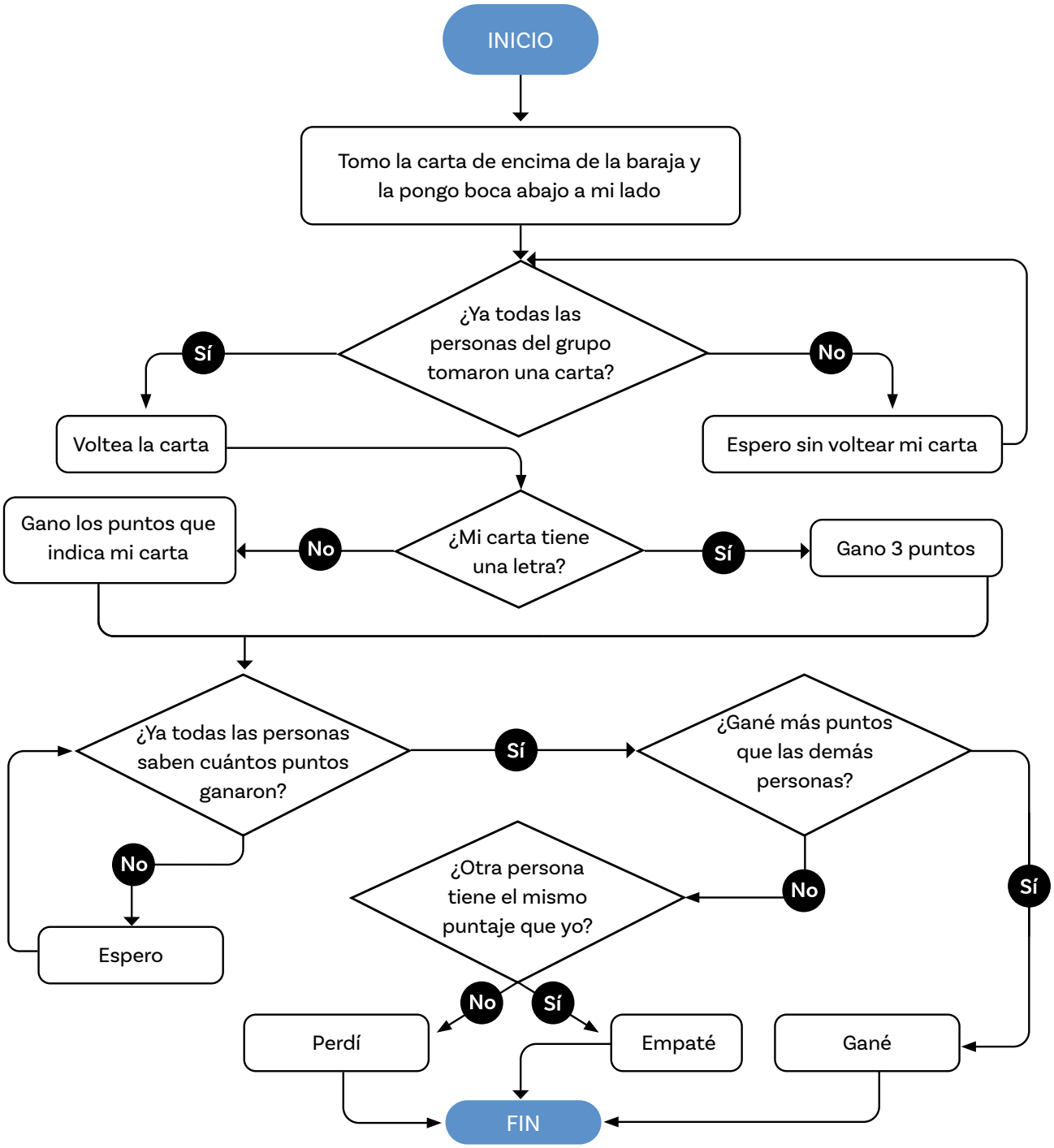
1. Debe haber 4 objetos en el escenario.
2. Cada objeto debe tener un solo color.
3. 3 objetos comienzan del mismo color y 1 de un color diferente. Este último representará a la lleva.
4. Los objetos se deben mover aleatoriamente y rebotar si tocan los bordes de la pantalla.
5. La lleva debe moverse más rápido que los demás objetos.
6. El primer cuadrante del escenario debe ser de otro color y corresponde a una zona segura o “tacho” en la que la lleva no puede congelar.
7. Cuando la lleva toca a otro objeto que no esté en el “tacho”, este se congela y cambia de color.
8. Cuando un objeto en movimiento, diferente de la lleva, toca a un objeto congelado, este se descongela.

Cuando tengan listo su programa, conviértanlo en un juego de computador en parejas. Con ese fin, cada jugador(a) debe hacer clic en iniciar y esperar hasta que la lleva congele los demás objetos en el escenario. Ganará la persona que logre que se congelen todos los objetos en el menor tiempo.



Anexo 1.2 Un algoritmo para jugar

Observen el siguiente diagrama de flujo.



Ahora, como grupo, van a probarlo. Necesitarán un mazo o baraja de cartas de póker.

Siéntense en círculo, barajen las cartas y póngalas en una pila, en el centro del grupo, boca abajo. Luego, sigan el algoritmo. En cada ronda deberá haber al menos una persona que gane o un empate. Jueguen varias rondas.

Luego, modifiquen el algoritmo, cambiando las reglas del juego. Por ejemplo, podrían pensar en una regla adicional para desempatar y asegurarse de que siempre haya una persona que gane. Hagan el diagrama de flujo correspondiente al juego que inventaron. Sigan las instrucciones de su docente para intercambiar diagramas de flujo con otro grupo y así ponerse a prueba jugando con nuevas reglas.



Anexo 2.1 Juego evalúa, decide y diviértete: tarjetas sorpresa

Tarjeta Sorpresa Si estás en el primer lugar en el tablero de juego, entonces te quedas en tu lugar. Si no, avanza hasta el lugar de la persona que vaya más adelantada en el tablero.	**Tarjeta Sorpresa** Si tu edad es mayor que la de las demás personas en el juego, entonces levántate y salta en un pie contando de uno en uno hasta que llegues a tu edad.	**Tarjeta Sorpresa** Si usas gafas o lentes de contacto, entonces las demás personas jugando deben darte un cumplido.	**Tarjeta Sorpresa** Si estás en el primer lugar en el tablero de juego y está lloviendo en este momento, entonces retrocede hasta el lugar de la persona que vaya más atrás en el tablero.
Tarjeta Sorpresa Si tienes ojos de color café y cabello crespo, entonces debes cantar una canción divertida al resto de jugadores y jugadoras.	**Tarjeta Sorpresa** Si tu edad es menor a 12 años, entonces cuéntales un chiste a las demás personas que estén jugando contigo.	**Tarjeta Sorpresa** Si todavía no es mediodía y tu edad no es mayor de 12 años, entonces avanza 2 casillas.	**Tarjeta Sorpresa** Si estás en la última posición del tablero, entonces avanza 3 casillas.
Tarjeta Sorpresa Si te gusta programar, entonces saca otra tarjeta sorpresa. Si no, haz 10 abdominales.	**Tarjeta Sorpresa** Si tu estatura es mayor que la de las demás personas en el juego, entonces dale una tarjeta sorpresa a la persona que sigue el turno después de ti.	**Tarjeta Sorpresa** Si estás en el último lugar del tablero de juego, entonces lanza el dado dos veces, suma los valores obtenidos en los dos lanzamientos y avanza ese número de casillas.	**Tarjeta Sorpresa** Si estás usando un saco o chaqueta, entonces póntelo al revés hasta el final del juego.
Tarjeta Sorpresa Si ya tomaste una tarjeta de sorpresa antes, entonces levántate e imita con tus movimientos a un animal de tu preferencia.	**Tarjeta Sorpresa** Si al lanzar el dado obtuviste un número par, canta una canción que tenga la palabra "vida". Si no logras cantarla, pierdes un turno.	**Tarjeta Sorpresa** Si está lloviendo y te gusta programar, entonces todas las demás personas que estén jugando contigo deben levantarse y hacer 5 sentadillas.	**Tarjeta Sorpresa** Si puedes decir correctamente 3 países con sus respectivas capitales, te quedas en tu lugar. Si no, retrocede hasta la anterior posición que ocupaste.
Tarjeta Sorpresa Si usas gafas y no estás usando tenis, entonces debes hacer un dibujo para las demás personas que estén jugando contigo.	**Tarjeta Sorpresa** Si al lanzar el dado obtuviste un número impar, entonces retrocede esa cantidad de casillas.	**Tarjeta Sorpresa** Si te gusta programar y estás aprendiendo muchas cosas al respecto, entonces date un aplauso y avanza 3 casillas más.	**Tarjeta Sorpresa** Si puedes decir correctamente 3 departamentos de Colombia y sus respectivas capitales, entonces puedes indicarle a otro jugador o jugadora que cuente un chiste.

Nota: adaptado del Juego de la lógica y las condiciones creado por ACOFI en 2021 para el programa Coding for Kids del Ministerio TIC, el MEN y el British Council en el marco del convenio 764 de 2021.

Anexo 2.2 Juego evalúa, decide y diviértete: tarjetas A

Tarjeta A: Está lloviendo en este momento.	**Tarjeta A:** Está haciendo sol en este momento.	**Tarjeta A:** Hoy es tu cumpleaños.	**Tarjeta A:** Usas gafas.
Tarjeta A: Llevas puesta alguna prenda de color blanco.	**Tarjeta A:** Tu edad es un número par.	**Tarjeta A:** Tienes ojos de color café.	**Tarjeta A:** Naciste en el mes de noviembre.
Tarjeta A: Escribes con la mano izquierda.	**Tarjeta A:** Llevas puesta alguna prenda de color verde.	**Tarjeta A:** Vives cerca de un río.	**Tarjeta A:** Hoy es viernes.
Tarjeta A: En tu escuela se separan los residuos entre reciclables y no reciclables.	**Tarjeta A:** Tu edad es un número impar.	**Tarjeta A:** Escribes con la mano derecha.	**Tarjeta A:** Naciste en el mes de mayo.
Tarjeta A: Hoy es miércoles.	**Tarjeta A:** Estás usando tenis.	**Tarjeta A:** Hoy es un día nublado.	**Tarjeta A:** Tienes dos nombres.
Tarjeta A: Naciste en un año par.	**Tarjeta A:** Naciste en un año impar.	**Tarjeta A:** Tienes una mascota.	**Tarjeta A:** Te gusta contar chistes.
Tarjeta A: Puedes decir los números del 1 al 10 en inglés.	**Tarjeta A:** Tienes el cabello corto.	**Tarjeta A:** No tienes mascotas.	**Tarjeta A:** Te gusta dibujar.
Tarjeta A: Puedes escribir con ambas manos.	**Tarjeta A:** Llevas puesta una prenda de color amarillo.	**Tarjeta A:** Llevas puesta una prenda de color café.	**Tarjeta A:** Tienes el cabello largo.

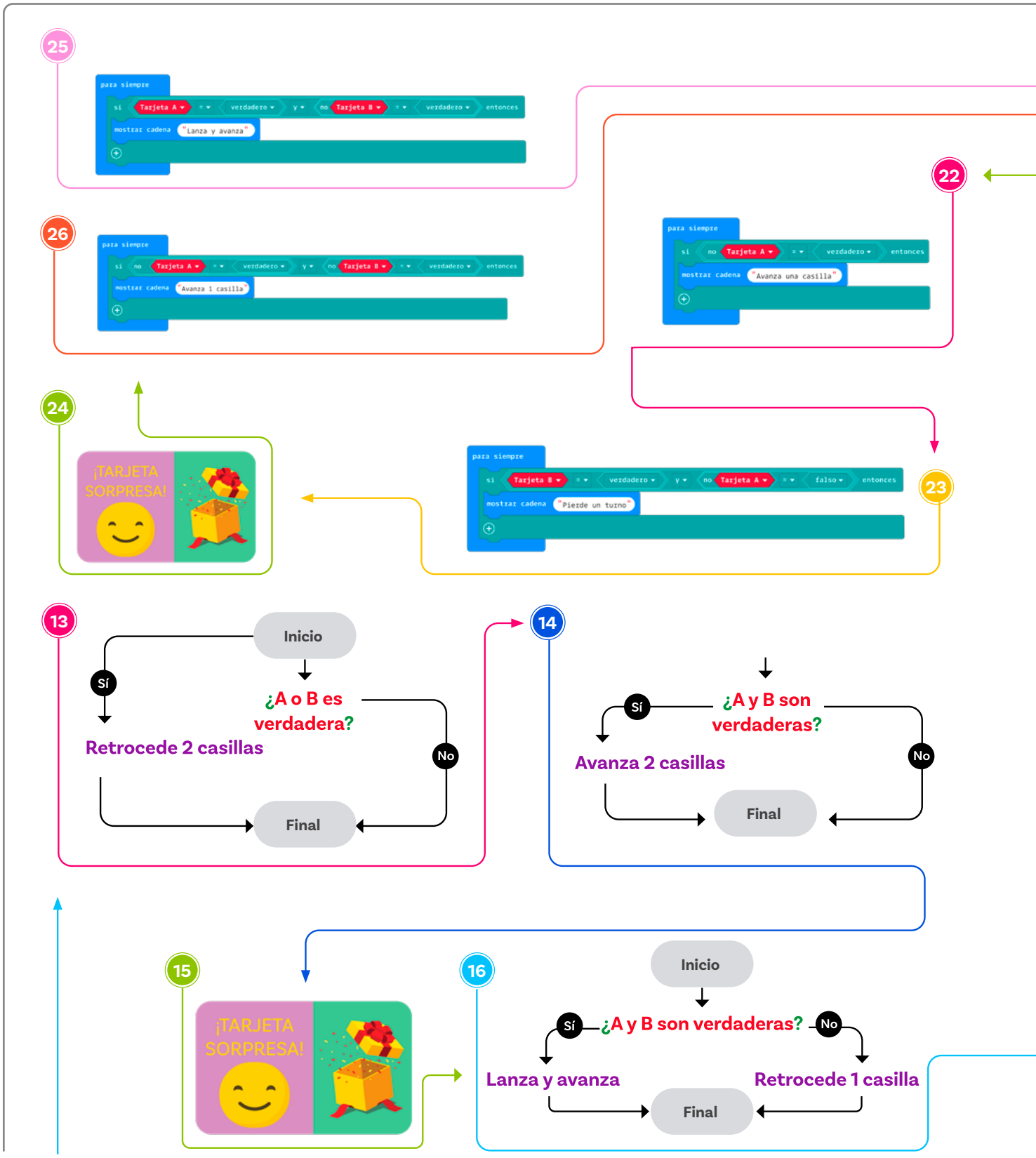
Nota: adaptado del Juego de la lógica y las condiciones creado por ACOFI en 2021 para el programa Coding for Kids del Ministerio TIC, el MEN y el British Council en el marco del convenio 764 de 2021.

Anexo 2.3 Juego evalúa, decide y diviértete: tarjetas B

Tarjeta B: Estudiantes en el salón de clases > 30	**Tarjeta B:** Tu edad < 13 años	**Tarjeta B:** Número de hermanos o hermanas = 2	**Tarjeta B:** Tu estatura > 1,4 m
Tarjeta B: Tu estatura < 1,4 m	**Tarjeta B:** Tu edad > 13 años	**Tarjeta B:** Vives en un municipio o ciudad a una altura sobre el nivel del mar > 1.000 m	**Tarjeta B:** Día del mes < 15
Tarjeta B: Suma edad de las personas en el equipo > 25	**Tarjeta B:** Suma edad de las personas en el equipo < 35	**Tarjeta B:** Día del mes > 15	**Tarjeta B:** Mes > 6
Tarjeta B: Promedio edad personas en el equipo < 13	**Tarjeta B:** Estudiantes en el salón de clases < 25	**Tarjeta B:** Talla de zapatos > 33	**Tarjeta B:** Doble de tu edad > 20
Tarjeta B: Talla de zapatos < 33	**Tarjeta B:** Número de hermanos o hermanas < 2	**Tarjeta B:** Mes < 6	**Tarjeta B:** Vives en un municipio o ciudad a una altura sobre el nivel del mar < 1.500 m
Tarjeta B: Hora del día > 9 am	**Tarjeta B:** Día del mes = 15	**Tarjeta B:** Promedio talla de zapatos de las personas en el equipo < 35	**Tarjeta B:** Mes = 6
Tarjeta B: Número de profesoras que te dan clases > 2	**Tarjeta B:** Número de profesores que te dan clases < 4	**Tarjeta B:** Personas del equipo a las que les gusta el color morado > 3	**Tarjeta B:** Personas del equipo a las que les gusta el color naranja < 3
Tarjeta B: Suma personas del equipo que están usando tenis > 3	**Tarjeta B:** Talla de zapatos = 33	**Tarjeta B:** Hora del día < 5 pm	**Tarjeta B:** Número de personas del salón de clases que usan gafas < 10

Nota: adaptado del Juego de la lógica y las condiciones creado por ACOFI en 2021 para el programa Coding for Kids del Ministerio TIC, el MEN y el British Council en el marco del convenio 764 de 2021.

Anexo 2.4 Evalúa, decide y diviértete: tablero de juego parte 1



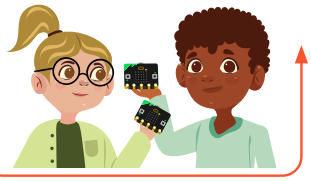
27



28

Si A no es verdadera y si B no es verdadera, entonces lanza y retrocede, si no avanza 1 casilla

Final



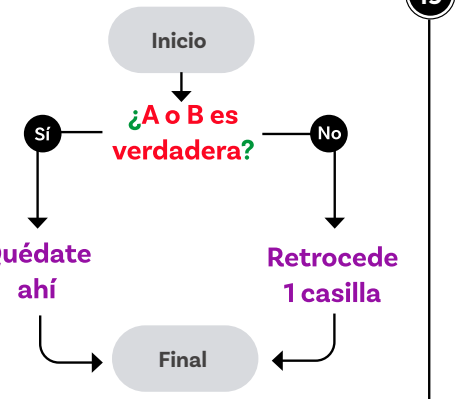
29



21



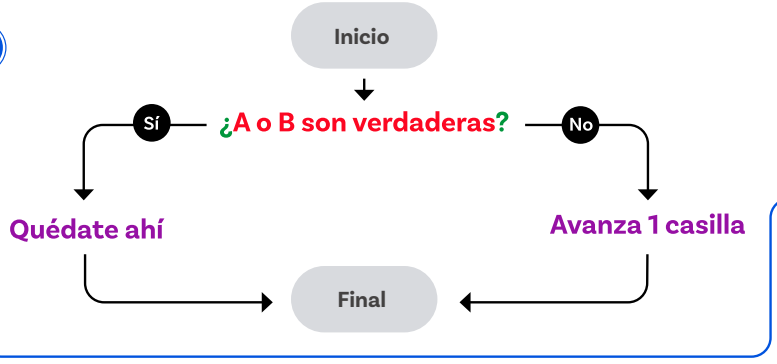
19



20



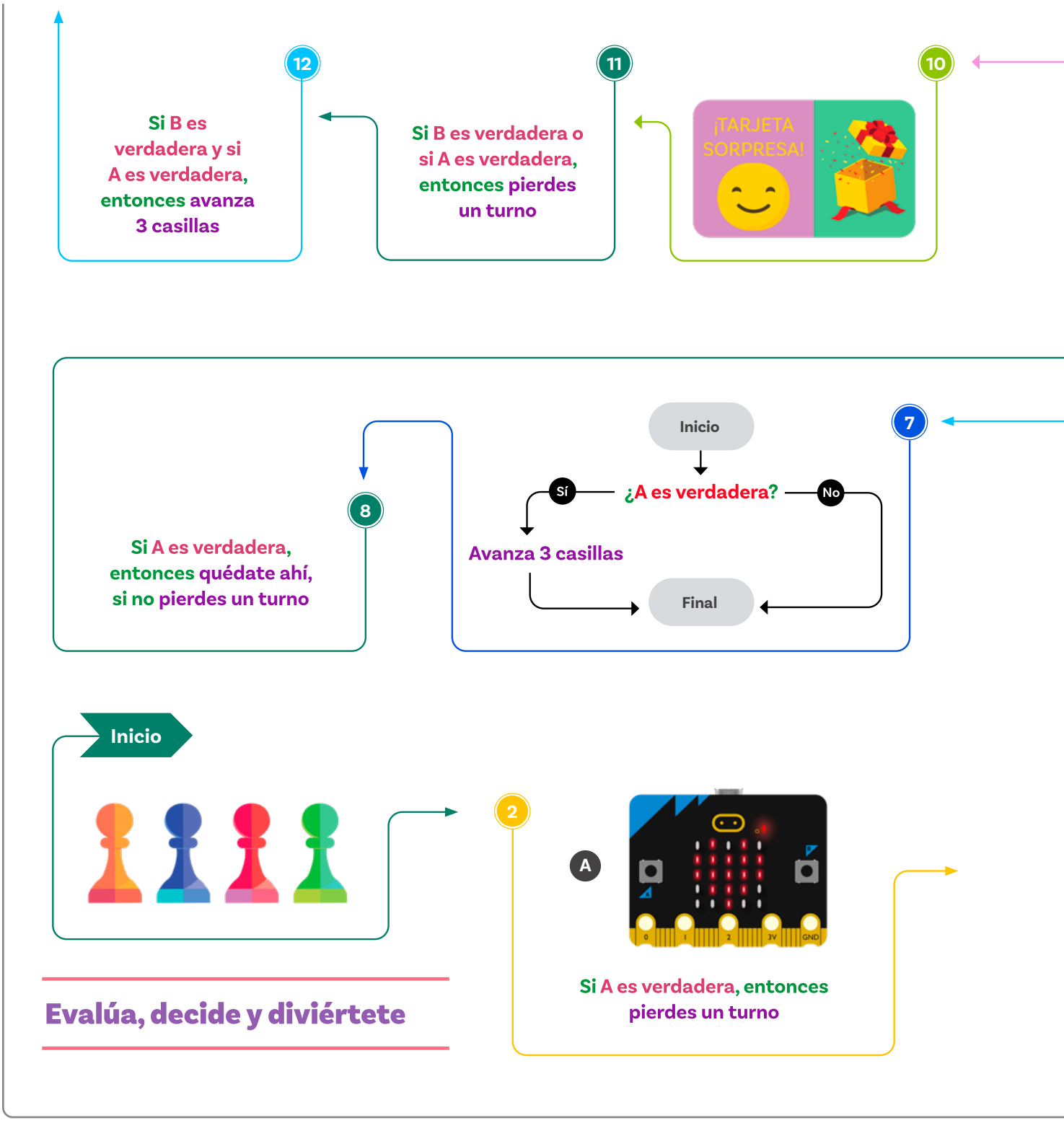
17



18

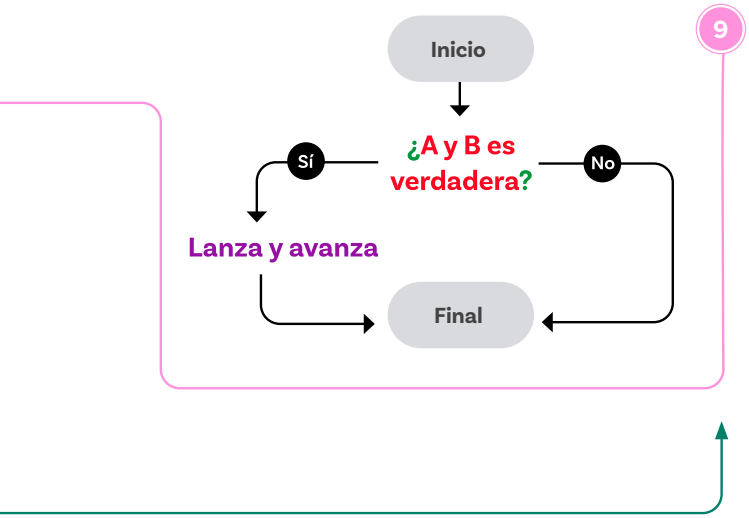


Anexo 2.4 Evalúa, decide y diviértete: tablero de juego parte 2



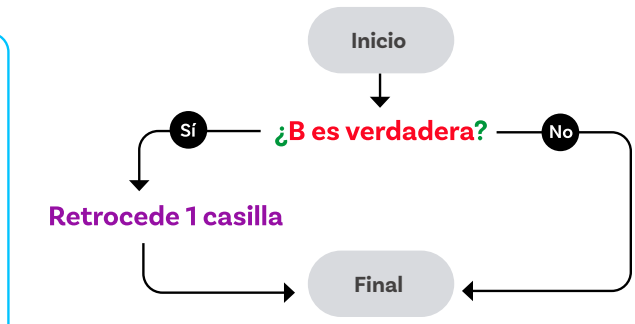
Nota: adaptado del Juego de la lógica y las condiciones creado por ACOFI en 2021 para el programa Coding for Kids del Ministerio TIC, el MEN y el British Council en el marco del convenio 764 de 2021.

9





6

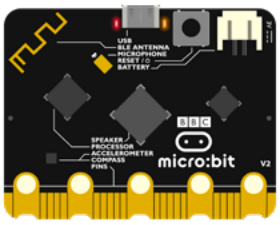


5

Si B es verdadera, entonces quédate ahí, si no pierdes un turno

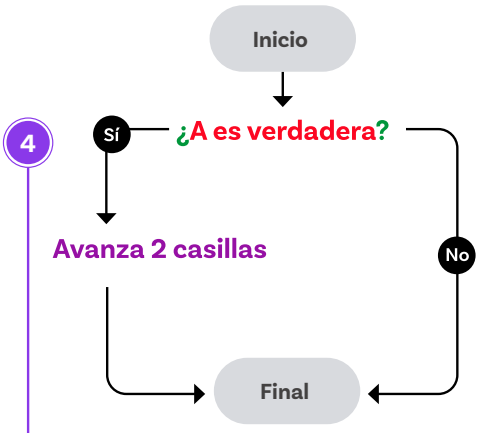
3

B

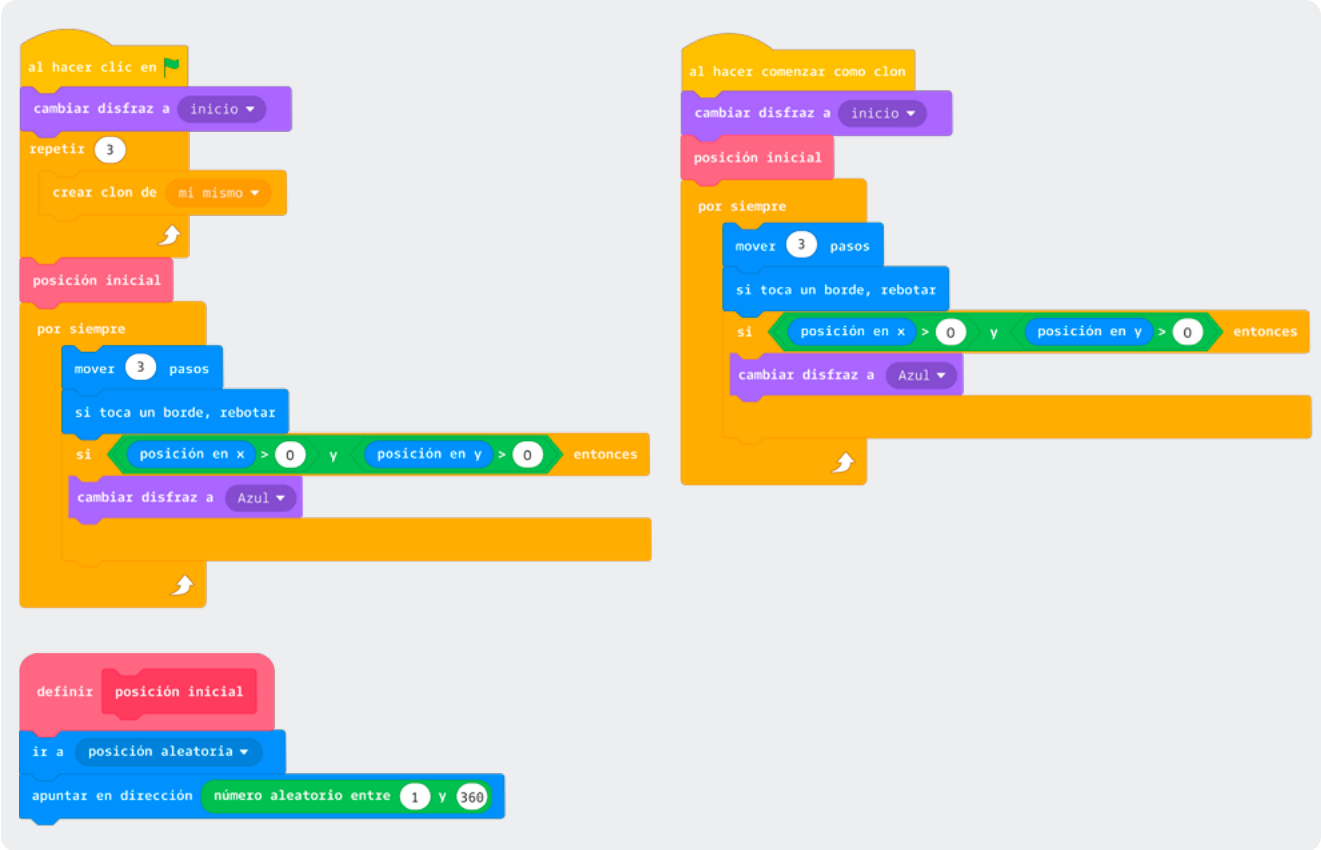


Si B es verdadera, entonces retrocede 2 casillas

4



Anexo 3.1 Código de partida



Anexo 4.1 Diagrama de flujo

Nombres: _____

Con respecto al algoritmo inicial, es decir, sin el condicional que está en blanco al principio, respondan:

1. Una vez se inicia este algoritmo, ¿cuántas veces se ejecuta?

_____

_____

2. ¿Cuáles son las primeras dos acciones que se realizan en este algoritmo?

_____

_____

3. ¿Dependen las primeras dos acciones de alguna condición?

_____

_____

4. ¿Deberían depender de alguna condición?

_____

_____

5. ¿De cuál? Recuerda la lógica booleana. Una condición puede consistir en **NO** ser algo. También una condición puede consistir en que dos expresiones sean verdaderas (**y**) o en que por lo menos una de las dos sea verdadera (**o**)

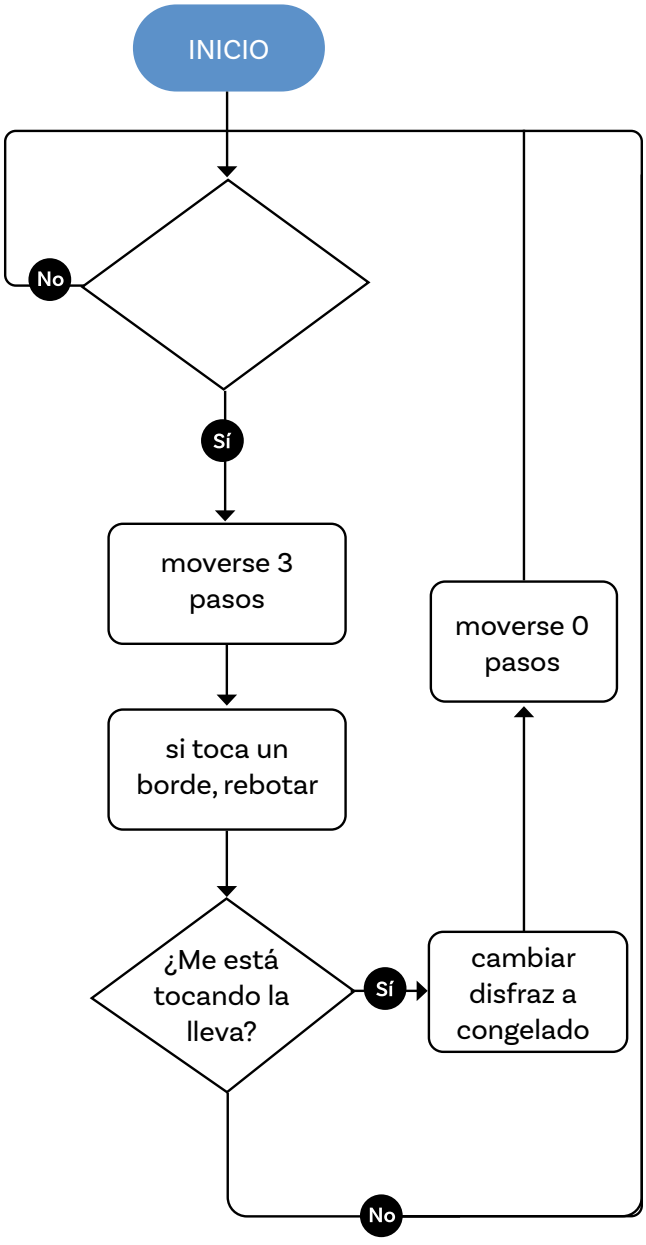
_____

_____

6. Escribe la condición en el condicional en el diagrama de flujo.

_____

_____



TIC

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}