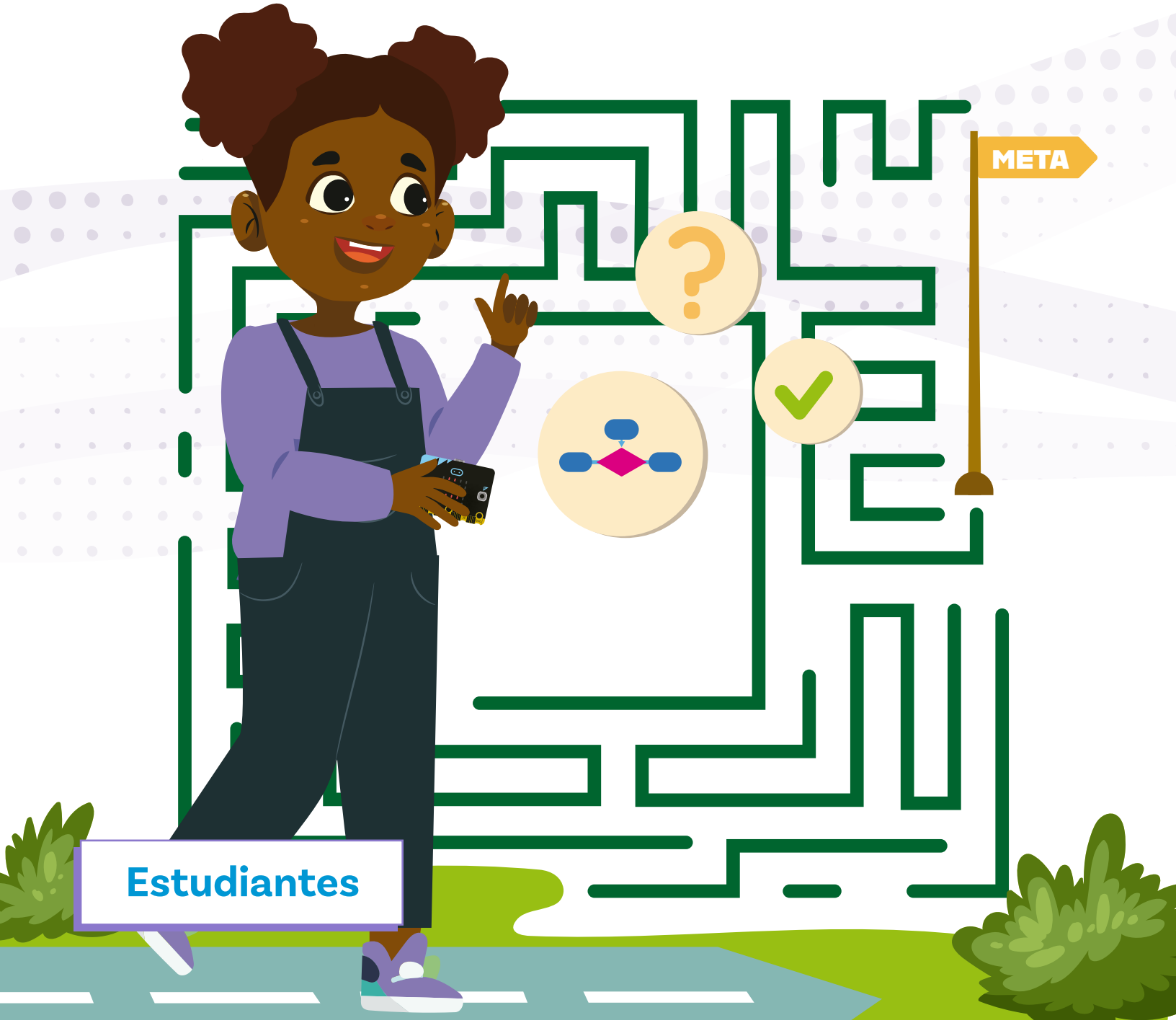
Saliendo de un laberinto con ayuda de un algoritmo: condicionales

Grado 5° Guía 3

TIC



Estudiantes

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

Saliendo de un laberinto con ayuda de un algoritmo: condicionales

Grado 5° Guía 3

TIC

Estudiantes

Apoya:

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

.Puntoaparte Editores

Diseño, diagramación, ilustración, y revisión de estilo

CC BY-NC 4.0

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: 978-628-7934-63-4

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.

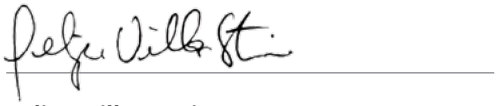
Encontrarán también en estas páginas orientaciones y sugerencias sobre cómo adaptar las actividades a diferentes contextos y necesidades. Por ejemplo, cómo desarrollar habilidades de pensamiento computacional en instituciones sin conectividad o incluso sin electricidad; cómo fomentar la equidad de género 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

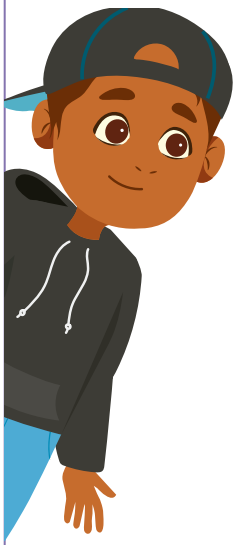
Creación

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.

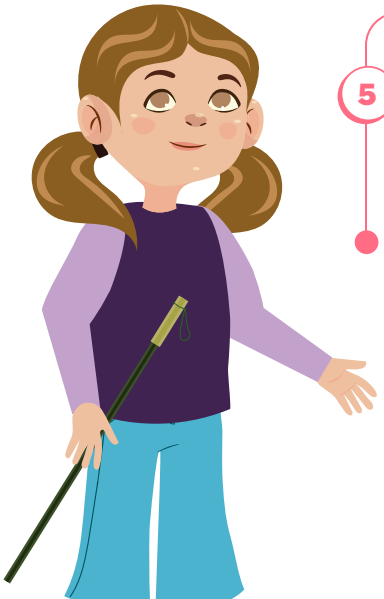
Mayo a diciembre 2024
Se incorporan a las guías los ajustes provenientes de las mesas de validación.

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 5° Guía 3

TIC

Identificación de los aprendizajes

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

Aprendizajes de la guía

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

- Crear diagramas de flujo para representar algoritmos con múltiples tomas de decisión.
- Usar entradas como los sensores de temperatura y luz y salidas como la pantalla de LEDs para recuperar o transmitir datos a un microprocesador.
- Utilizar condicionales y operaciones lógicas para decidir qué acciones ejecutar.

Resumen de la guía

Esta guía propone 5 sesiones de trabajo dirigidas al aprendizaje y apropiación de los condicionales como estructuras para controlar la ejecución de algoritmos.

En la guía se presentan actividades desconectadas y conectadas, incluyendo el uso de diferentes variables de entrada de la *micro:bit* como los botones, sensores de luz y temperatura y el acelerómetro.

Resumen de las sesiones

Sesión 1	Se presenta el concepto de condicionales a través de actividades desconectadas en relación con las reglas y condiciones que se cumplen en el día a día.
Sesión 2	Se aprende a usar condicionales para controlar programas sencillos en *MakeCode*.
Sesión 3	A través de una actividad desconectada para salir de laberintos, se implementa un algoritmo que incluye condicionales para generalizar la solución de un problema.

Aprendizajes de la guía

- Analizar programas o algoritmos sencillos y determinar si la salida o resultado que producen es el esperado.
- Crear y depurar programas que incorporen tomas de decisión a partir de la comparación de variables booleanas o numéricas.

Sesión 4

Se presentan los condicionales de varias condiciones de tipo “si”, “si no, si”, “si no” y se utilizan para programar un sistema de direccionales.

Sesión 5

Se utilizan los operadores lógicos “o” e “y”, para unir condiciones. Se practica su uso con actividades desconectadas y conectadas.

Conexión con otras áreas

A continuación, se presenta la conexión con otras áreas:

Matemáticas

- En las sesiones 1, 2 y 3 se aborda el manejo de variables aleatorias lo que se podrá conectar en matemáticas con el pensamiento aleatorio.

Ciencias naturales

- En las sesiones 1, 2 y 3 se aborda la simulación en la evolución de la población de dos especies en un modelo sencillo de predador-presa. Este aprendizaje ayudará a este tipo de aprendizajes en ciencias naturales posteriormente.

Nota

Esta guía es una adaptación de la ficha “Expedición espeleológica”, desarrollada en el 2020 por ACOFI para el programa *Coding for Kids*, en el marco del convenio 838 entre el Ministerio TIC, Computadores para Educar y el British Council, y editada en el marco de los convenios 764 de 2021 y 698/002 de 2022 suscritos entre el MEN, el Ministerio TIC y el British Council.

Grado 5° Guía 3

TIC

Sesión 1

Aprendizajes esperados

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

- Reconocer que existen condiciones que definen si se ejecutan o no algunas acciones de un algoritmo.
- Determinar si una condición se cumple.

Material para la sesión

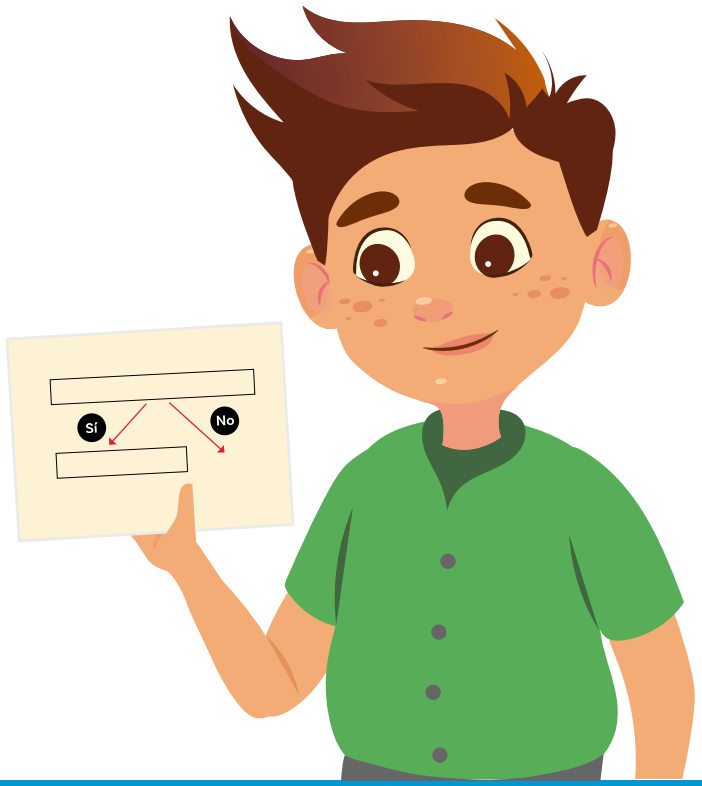
Material por grupo:

- Un dado.

Duración sugerida



40% 40% 20%



Lo que sabemos, lo que debemos saber



Esta sección corresponde al 40% de avance de la sesión

La guía anterior te ayudó a ampliar tus conocimientos sobre los ciclos o bucles y te permitió reconocer algunas situaciones en las que se deben repetir acciones. Como notaste, hubo casos en los que se generaban bucles si se cumplía alguna condición. Esta guía te ayudará a aprender más acerca de otras formas de condicionales.

Para empezar, analiza algunos ejemplos:

Tal vez te ha pasado que tu docente dice:



Si todo el salón hace silencio por los próximos 20 minutos, entonces salimos 5 minutos antes al descanso.



¿Y qué pasa en estos casos?

¿Qué crees que pasa si todo el salón logra hacer silencio por 20 minutos?

¿Pero... qué pasa si alguien habla antes de que acabe el tiempo?

Normalmente, lo que pasa es que, si todo el salón logra cumplir la meta y estar en silencio, entonces tu docente cumple su promesa y todos salen a descanso cinco minutos antes de la hora normal. Pero, si alguna de tus compañeras o compañeros habla antes de que acabe el tiempo, no cumplen la meta y no pueden disfrutar de cinco minutos extra de descanso.

Lo que pasa es que tu docente pone una **condición**. Y solo cuando se cumple esa condición, pueden ver la recompensa.

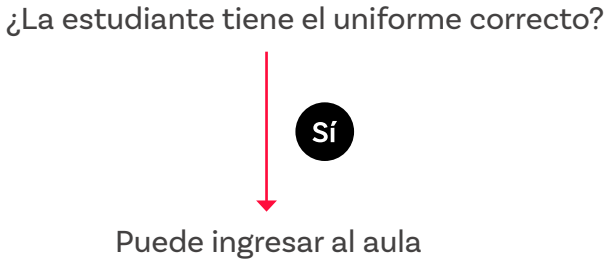
En la vida cotidiana también vemos muchas condiciones o reglas. Por ejemplo, en tu casa puede que tengas una hora destinada para ir a dormir y en el colegio debes usar un uniforme.

Una forma de escribir esas reglas puede ser:

- Si una estudiante tiene el uniforme correcto, entonces puede entrar al colegio.
- Si un estudiante completa todas las tareas, entonces obtendrá una buena calificación en la clase.
- Si recoges tus juguetes después de jugar, entonces mantendrás tu habitación ordenada.

En un diagrama de flujo, podría verse como en la *Figura 1.*

Figura 1. Diagrama de flujo



Esas reglas sirven como condiciones que debemos cumplir para poder hacer otras cosas. ¿Puedes pensar en otras condiciones parecidas?

Escribe dos ejemplos aquí y compártelos con alguien más de tu salón.

1. _____

_____

_____

2. _____

_____

_____

En las próximas sesiones vas a aprender más sobre esas **condiciones** que se deben cumplir para que sucedan otras cosas. Y, más adelante, verás cómo se pueden controlar los programas también con sus propias condiciones.

Glosario

- **Condición:** es una frase que puede ser verdadera o falsa. Si es verdadera, se hacen las acciones. Si es falsa, no se hacen.
- **Si... entonces:** es es la forma en que se escriben las condiciones. La parte de "Si" indica la condición y la parte de "entonces" indica lo que se debe hacer, es decir, las acciones que deben realizarse.



Manos a la obra
Desconectadas

Esta sección corresponde al 80% de avance de la sesión

De forma individual, resuelve las siguientes actividades. Después, sigue las instrucciones de tu docente para comparar tus respuestas con las de otra compañera o compañero antes de que se revisen las respuestas con toda la clase.

Actividad 1:
Los siguientes retos tienen condiciones. Sigue el ejemplo y completa las frases.

1. Si levantas la mano, entonces puedes participar en clase.
2. Si parece que va a llover, entonces _____
3. Si al dividir un número por 2 el residuo es 0, entonces

_____

4 Si el reloj _____, entonces se puede salir al recreo.

5 Si el árbol recibe suficiente agua, entonces _____

Actividad 2:
La siguiente tabla contiene información sobre cuatro personas diferentes. Léela con atención, pues necesitarás estos datos para poder desarrollar los ejercicios propuestos después. En la última fila debes agregar tu nombre e información (ciudad de residencia, edad, mascota).

Tabla 1. Información actividad 2

Nombre	Información
Ana	Vive en Barranquilla, tiene 17 años, su mascota es un pez dorado.
Mario	Vive en Itagüí, tiene 18 años, su mascota es un loro.
Carlos	Vive en Leticia, tiene 52 años, no tiene mascota.
Matilde	Vive en Pasto, tiene 43 años, su mascota es un gato.
Tu nombre:	Tu información:

Ahora, lee las situaciones que se plantean a continuación y sigue el diagrama de flujo para determinar cuáles de las personas de la tabla anterior las cumplen.

a. En Colombia, la edad legal para votar es 18 años. Completa la tabla indicando quiénes pueden votar ya y quienes todavía no pueden hacerlo.

Figura 2. Diagrama de flujo para decidir quién puede votar

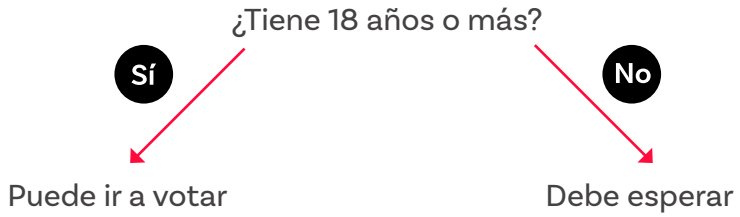


Tabla 2. ¿Puede votar?

Nombre	¿Puede votar?
Ana	No
Mario	
Carlos	
Matilde	

b. En Colombia, las personas pueden obtener su licencia de conducción a partir de los 16 años. Completa la tabla indicando quienes podrían obtener una licencia de conducción.

Tabla 3. ¿Podría conseguir una licencia?

Dibuja el diagrama de flujo	Nombre	¿Podría conseguir una licencia para conducir?
	Ana	
	Mario	
	Carlos	
	Matilde	

c. Una tienda veterinaria quiere ofrecer un servicio de peluquería para las mascotas. Completa el diagrama de flujo en la *Figura 3* y utilízalo para decidir quiénes podrían obtener este servicio.

Figura 3. Diagrama de flujo servicio de peluquería para mascotas

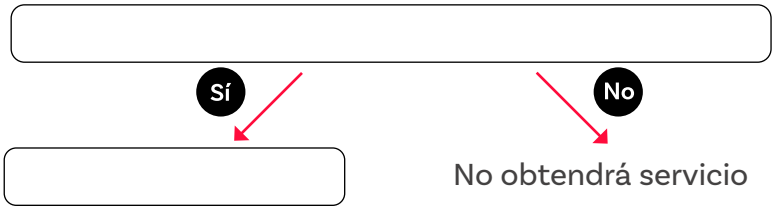


Tabla 4. ¿Obtendrá el servicio de peluquería?

Nombre	¿Obtendrá el servicio de peluquería de mascotas?
Ana	No
Mario	
Carlos	
Matilde	

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes previstos para esta sección ¿Los lograste? Responde en un cuaderno a las siguientes preguntas:

¿Se podrían hacer programas sin usar ninguna condición? ¿Se te viene a la mente algún ejemplo?

Ahora puedes:

- ○ Reconocer que existen condiciones que definen si se ejecutan o no algunas acciones.
- ○ Determinar si una condición se cumple o no.

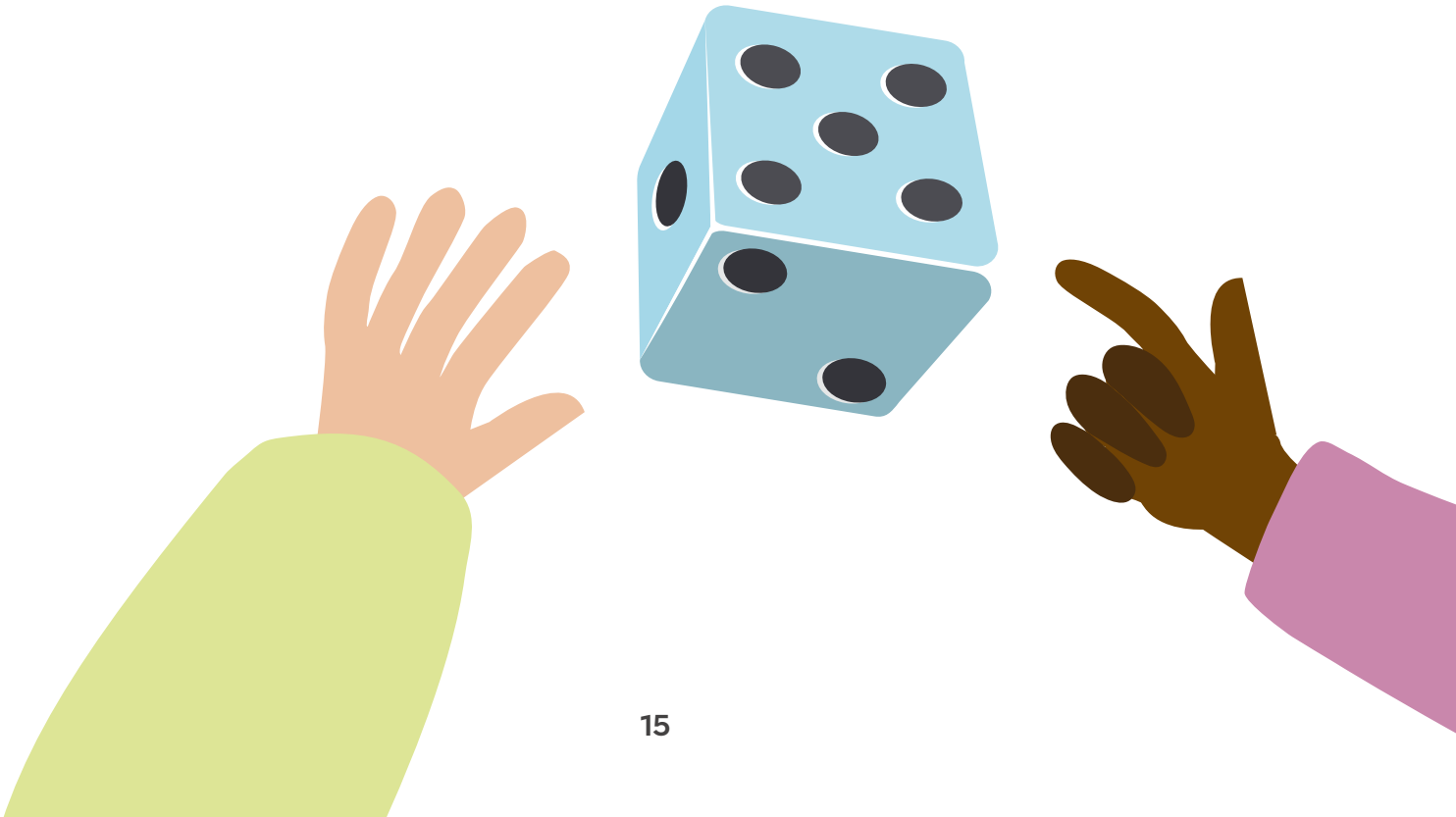
Actividad opcional para seguir consolidando aprendizajes

Ya que has aprendido a utilizar condiciones, ¿qué tal si las usas para crear el algoritmo de un pequeño juego de penitencias?

1. Trabaja con una compañera o compañero. Creen condiciones que deban cumplir otras personas de su clase si sacan un número en particular al tirar un dado.

2. En sus cuadernos, escriban mínimo 6 condiciones. Por ejemplo, podrían escribir algo como esto:
   Si sacan 1, entonces cuéntennos un chiste.
   Si sacan 2, entonces hagan 3 sentadillas.

3. Cuando hayan terminado, sigan las instrucciones de su docente para trabajar con otros grupos. Tomen turnos para lanzar un dado y cumplir las condiciones propuestas por sus compañeras y compañeros.



4 Al finalizar discutan las siguientes preguntas:

¿?

¿Qué grupos propusieron las mejores condiciones?

¿Hubo alguna condición que no pudieran o quisieran cumplir? ¿Por qué?

¿Hubo alguna condición que escribieron y que nadie tuvo que cumplir? ¿Por qué?

¿Creen que se podría hacer un juego computarizado que use condiciones como en el juego que crearon?

Revisa los aprendizajes de la sesión. ¿Crees que lograste alcanzarlos?

1 ¿Puedes reconocer que existen condiciones que definen si se ejecutan o no algunas acciones?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2 ¿Puedes determinar si una condición se cumple o no?

- ◯ Sí
- ◯ Parcialmente
- ◯ No

Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", piensa nuevamente en el juego. Cada condición que escribieron representaba una acción que otras personas debían ejecutar. Por eso, si al arrojar el dado se obtenía un número en particular, se debía ejecutar esa acción. Pero si un número no salía en los dados, la condición asociada a ese número no se cumplía. Si sientes que todavía tienes alguna duda sobre este tema, pide ayuda a tu docente.

Grado 5º | Guía 3

TIC

Sesión 2

Aprendizajes esperados

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

- Crear programas en *MakeCode* con tomas de decisiones sencillas basadas en la lectura y comparación de variables numéricas.
- Plantear algoritmos que incorporen el uso de variables físicas como la temperatura y los niveles de luz, leídas mediante sensores.
- Determinar si la salida o resultado de un algoritmo es el esperado, según condiciones indicadas.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Dispositivo con acceso a *MakeCode*.

Nota

Recuerda que *MakeCode* cuenta con herramientas de accesibilidad como alto contraste y funciona con lectores en pantalla. Pide apoyo a tu docente si necesitas usar alguna de estas opciones.

Lo que sabemos, lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

En la sesión pasada aprendiste sobre las condiciones y descubriste que hay tareas o actividades que solo se ejecutan si se cumple una condición (si esa condición es VERDADERA). Para practicar, vas a jugar un juego.

Este juego se puede jugar por grupos, o puede ser una dinámica de todo el salón, según lo indique tu docente. Una persona será la líder del juego y las demás deberán hacerse en filas, en un extremo del salón, pero mirando hacia al frente.

La persona que lidera el juego será la programadora de condicionales. Esta persona elige una condición y les pide a todas y todos los que cumplan con esa condición, que den un paso hacia adelante.

- Si tienes el pelo negro, entonces da un paso hacia adelante.
- Si tu nombre empieza por A, entonces da un paso hacia adelante.

Tú y el resto del grupo deben evaluar esa condición y decidir si caminan o no hacia adelante.

También pueden intentar con condiciones negativas como:

- Si NO tienes el cabello corto, entonces da un paso adelante.
- Si tu cumpleaños NO es en abril, entonces da un paso adelante.

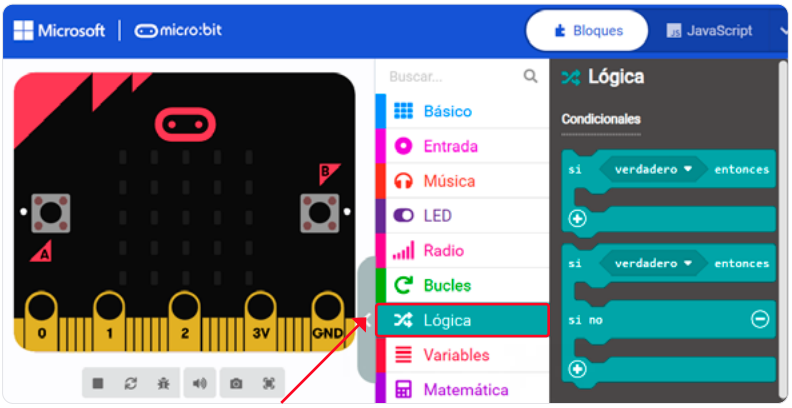
Cuando al menos tres personas hayan llegado hasta el otro extremo del salón, cambien de roles y vuelvan a empezar.

¿Cómo te fue evaluando condicionales? ¿Te confundiste en algún momento? Lo importante de este juego es que recuerdes que las **condiciones son una forma de controlar las acciones que se realizan**.

Cuando se programa también se debe controlar el código con algunas condiciones y, para eso, se usan los **condicionales**.

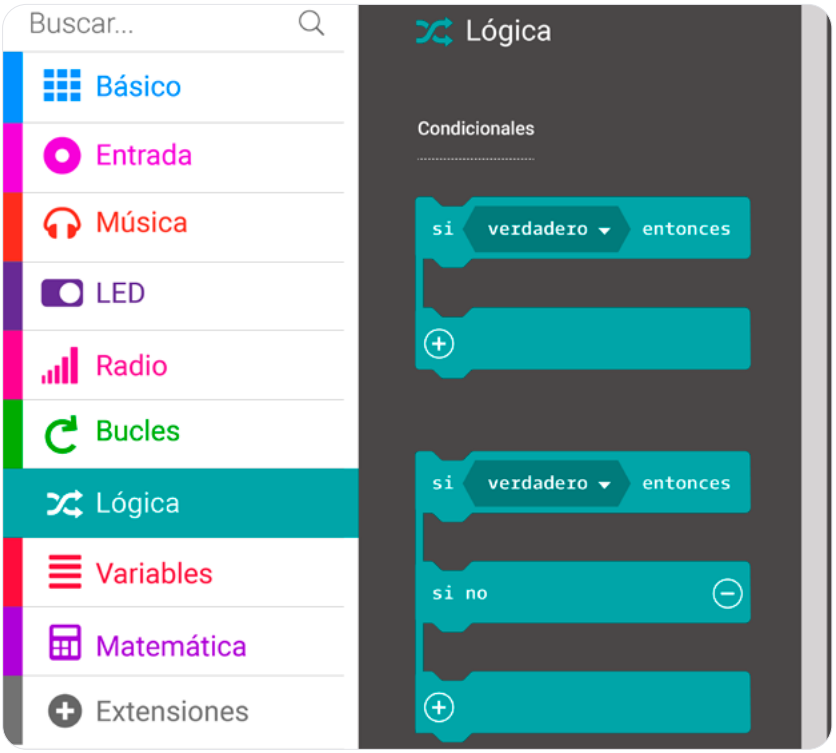
Los bloques condicionales de *MakeCode* se encuentran en la caja de herramientas, en el menú Lógica.

Figura 1. Menú Lógica con condicionales en *MakeCode*



Estos bloques se ven así:

Figura 2. Bloques condicionales de *MakeCode*



Nota

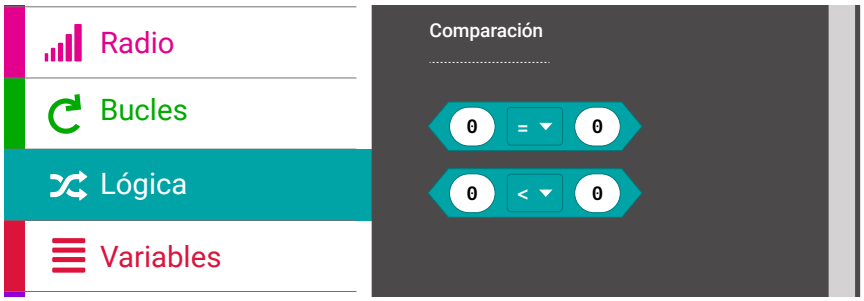
Los LEDs y los botones son variables booleanas pues pueden asumir dos valores solamente: verdadero o falso.

Por eso cuando el botón A está oprimido, su valor es verdadero y, cuando no, es falso.

Como puedes ver, tienen la misma estructura que ya has practicado: Si (algo es verdadero) **entonces**...

Además, se pueden combinar con los bloques de comparación que se ven, como se muestra en la *Figura 3*.

Figura 3. Bloques de comparación en *MakeCode*

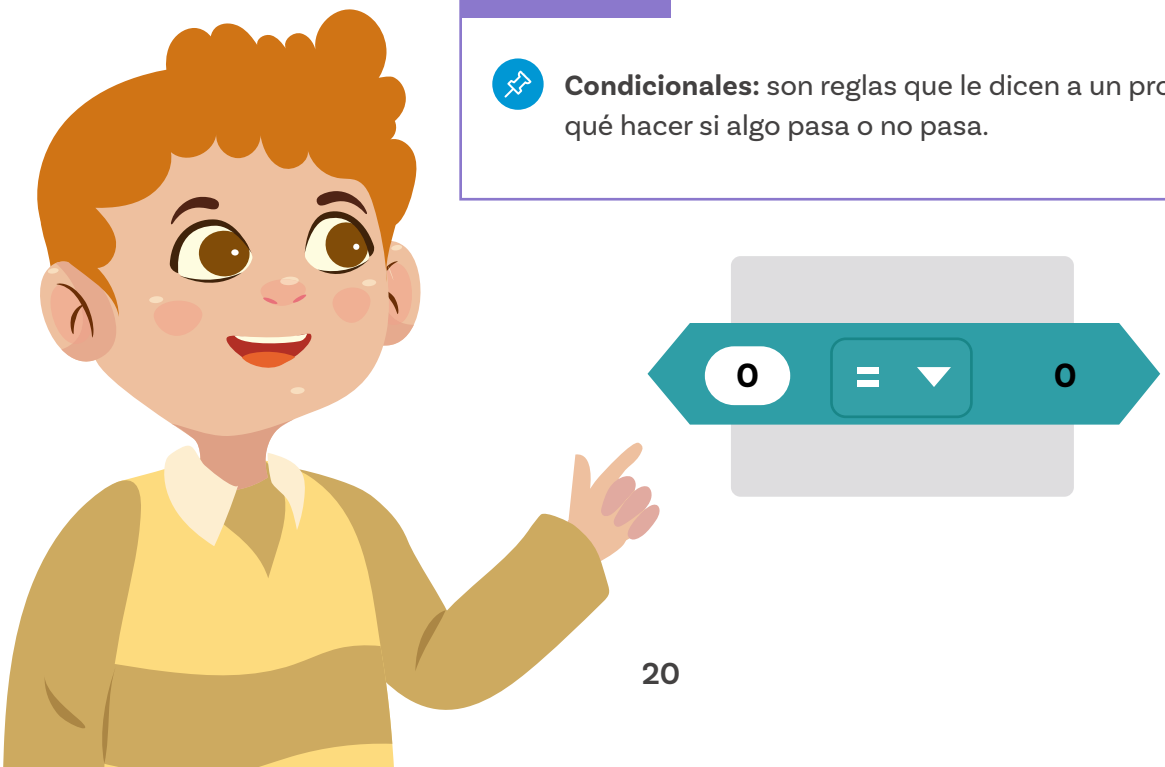


Al utilizar estos bloques puedes comparar si dos valores son iguales (=), o si alguno de los valores es menor que (<) o mayor que (>) otro.

Con estos nuevos bloques podrás diseñar programas cada vez más complejos e interesantes. En la siguiente sección vas a aprender a usar los bloques condicionales en *MakeCode*, utilizando ejercicios prácticos.

Glosario

Condicionales: son reglas que le dicen a un programa qué hacer si algo pasa o no pasa.



Enlace

Recuerda que puedes acceder al editor *MakeCode* en:

Manos a la obra

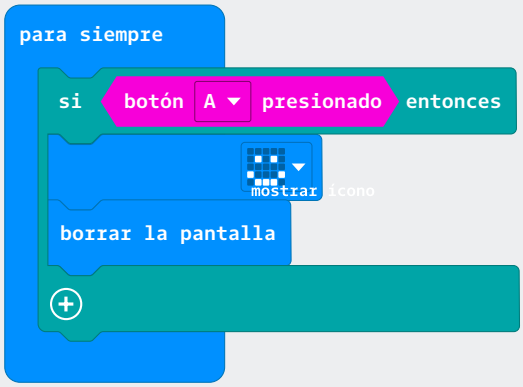
Conectadas

Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos de 2 o 3 personas, siguiendo las instrucciones de tu docente.

Actividad 1: trabajando con botones

Observen el siguiente código. ¿Qué creen que hace? Escriban una explicación corta de lo que creen que pasará al ejecutarlo.



¿Para qué creen que se incluye la instrucción borrar la pantalla?

¿Qué sucede si no se incluye el bloque borrar la pantalla?

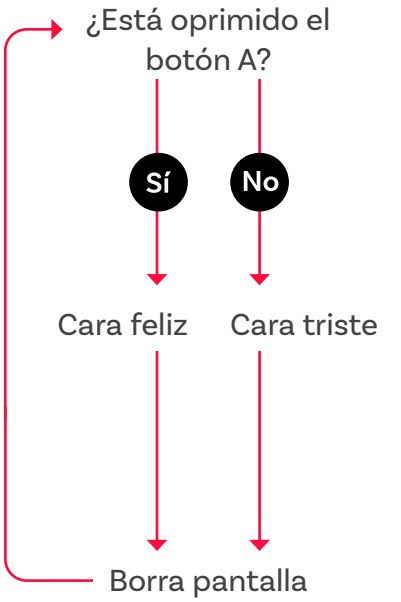
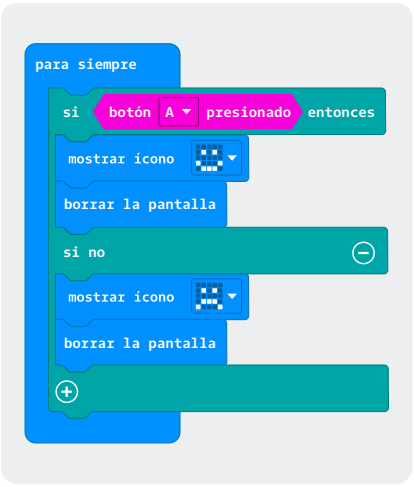


Figura 1. Código con condicional Si entonces, Si no



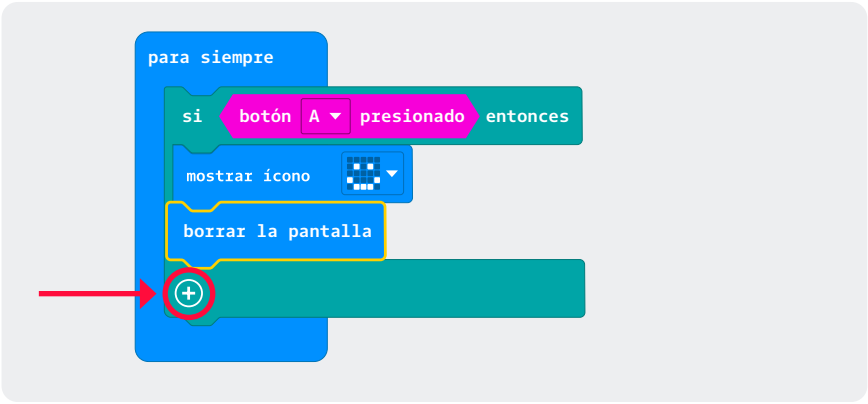
Ahora, ingresen a *MakeCode* creen el programa y pongan a prueba su predicción.

Examinen ahora el diagrama de flujo que aparece en la izquierda. ¿Qué creen que hace?

¿En qué se diferencia del diagrama de bloques que acaban de observar?

Como probablemente hayan notado, el diagrama de flujo tiene ahora dos caminos: uno que dice qué hacer **si** se cumple la condición y otro que dice qué hacer **si no** se cumple la condición.

Ahora, hagan clic en el signo (+) que aparece en la parte de abajo a la izquierda del bloque si... entonces. Esto les permitirá añadir la opción si no.



Completen el código en *MakeCode* para que la tarjeta *micro:bit* muestre una cara feliz si el botón A está presionado, y muestre una cara triste si no está presionado.

Verifiquen que su código es igual al de la *Figura 1*.

En este caso, al no estar oprimido el botón A, se verá una cara triste.

> *¿Se requiere en este caso incluir el bloque borrar pantalla? ¿Por qué sí o por qué no?*

Verifiquen sus respuestas en el editor.

Ahora modifiquen su código para que siga la siguiente condición: Si el botón B está presionado, entonces debe mostrarse un signo de verificación o chulo ✓, si no, se debe mostrar una X.

Actividad 2: trabajando con sensores

En la guía anterior trabajaron con unas entradas de la *micro:bit* llamadas sensores, usando específicamente el sensor de temperatura. Ahora van a utilizar los sensores de temperatura y de luz para controlar sus programas.

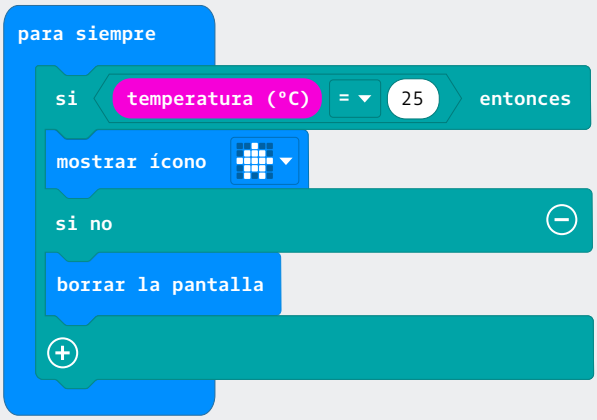
Piensen en la siguiente situación: Lucía y Santiago quieren programar la *micro:bit* para que genere una alarma que le indique a su hijo Alejandro y a su hija Lorena que ya es momento de entrar a la casa. Ambos quieren que la alarma cumpla con las siguientes condiciones:

- Si está haciendo frío, entonces la alarma debe mostrar el dibujo de una casa.
- Si se hace oscuro, pues está haciéndose de noche, la alarma también debe mostrar una casa.
- Si no se cumple ninguna de las dos condiciones, la alarma no debe mostrar nada.

Si tienen alguna dificultad creando su programa, pidan permiso a su docente para levantarse silenciosamente y observar lo que otros grupos están programando. Esto les permitirá notar diferencias e inspirarse para regresar y ajustar su código.



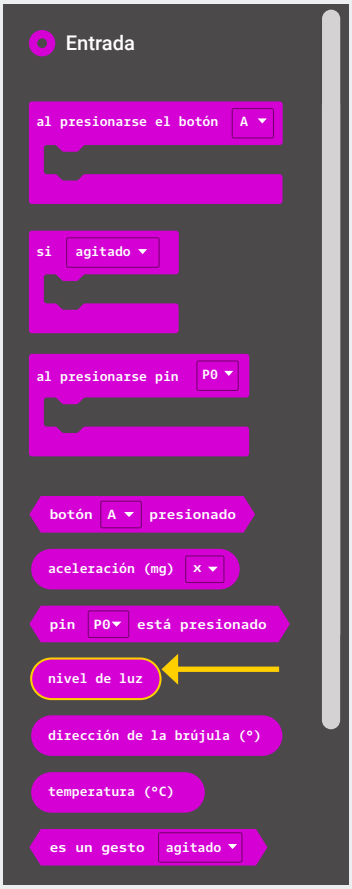
Lucía siente que cualquier temperatura de menos de 25°C es muy fría para que Alejandro y Lorena estén fuera de casa jugando, y aunque ella misma programó la *micro:bit* con el fin de generar la alerta de temperatura, notó que esta parte de la alarma no está funcionando como ella y su esposo esperaban.



¿Pueden identificar en dónde está el error o errores en el código que hizo Lucía?

Corrijan el código en el editor.

Figura 2. Ubicación del bloque para usar el sensor de luz



Una vez hayan finalizado esta parte de la alarma, completen el programa. Tengan en cuenta que Santiago piensa que si el nivel de luz que detecta la *micro:bit* es menor a 100, ya está muy oscuro para que su hijo e hija estén fuera de casa. La *Figura 2* les muestra dónde encontrar el bloque requerido para que su programa haga uso del sensor de luz.

Cuando programan, ustedes también toman decisiones en el diseño. Tengan en cuenta que no existen soluciones perfectas y que se puede llegar de muchas maneras al mismo objetivo.

Ahora es su turno de crear sus propios programas utilizando **condicionales**. A continuación, se presentan dos situaciones en las que se debe crear un programa.

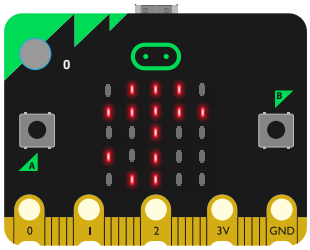
Situación 1
Creen un programa que muestre una cara feliz **si** la temperatura es mayor a 30 grados y que muestre una cara triste **si no** es mayor a 30 grados.

Situación 2
Si van a salir y observan que **está muy oscuro**, saben que es posible que llueva y que podrían necesitar un paraguas. Pero **si está soleado**, piensan que quizás es mejor llevar gafas de sol.

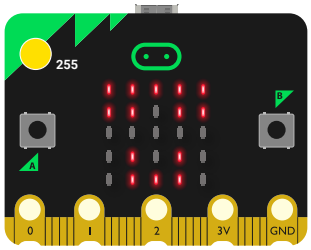
Creen un programa que decida si deberían llevar paraguas o gafas de sol para salir.

Verifiquen que su programa de la situación 2 esté bien.

La imagen del paraguas se ve así:



La imagen de las gafas se ve así:



Cuando terminen sus programas, compárenlos con los de otro grupo y luego respondan a las siguientes preguntas:

¿Son los programas de los dos grupos totalmente iguales? ¿En qué se parecen y en qué se diferencian?

¿Cuántos bloques usaron para lograr su objetivo? ¿Podrían utilizar menos bloques?



Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes de la sesión. ¿Crees que lograste alcanzarlos?

1. ¿Puedes crear programas en *MakeCode* con tomas de decisiones sencillas basadas en la lectura y comparación de variables numéricas?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes plantear algoritmos que incorporen el uso de variables físicas como la temperatura y los niveles de luz, leídas mediante sensores?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

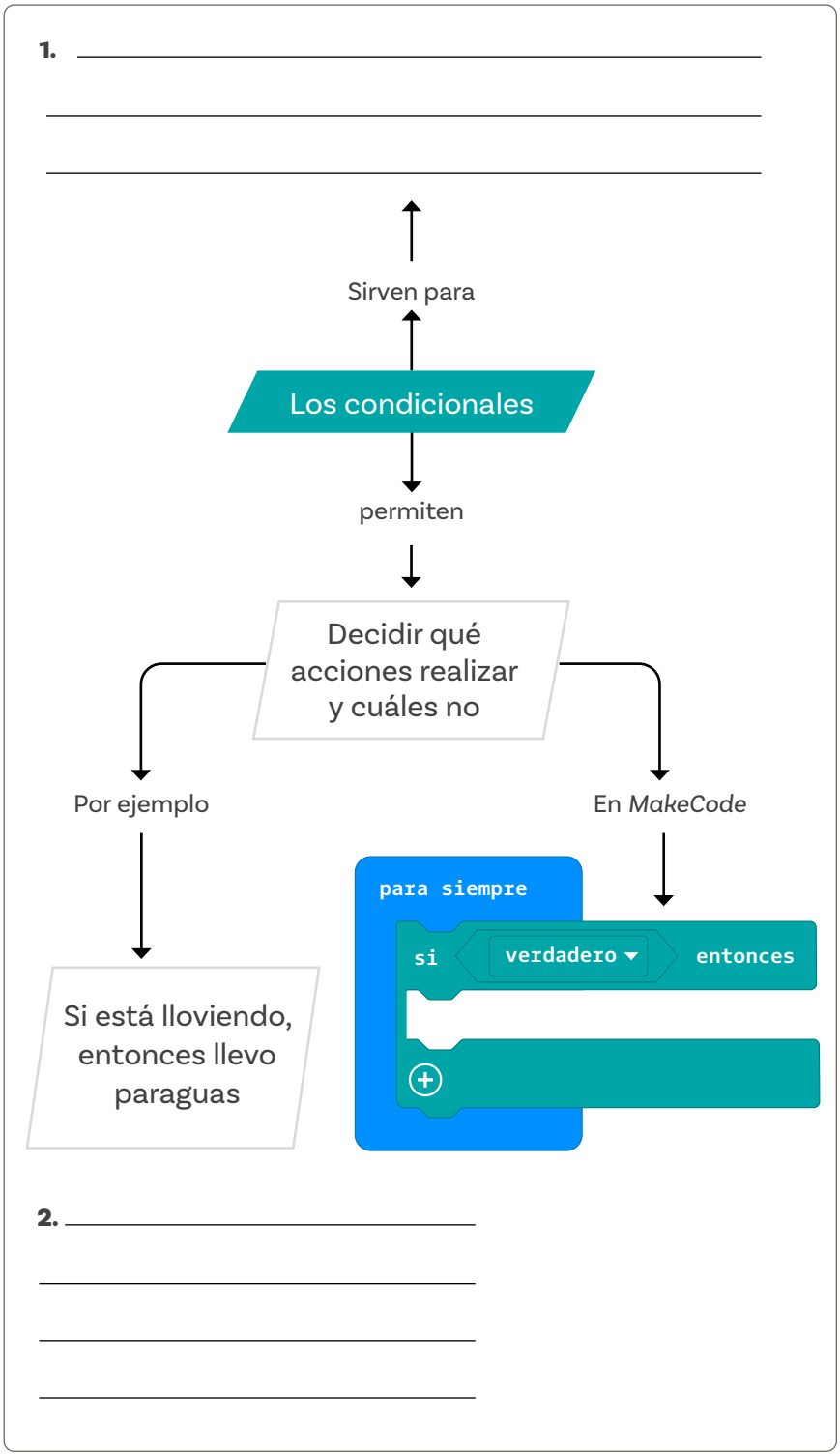
3. ¿Puedes determinar si la salida o resultado de un algoritmo es el esperado, según condiciones dadas?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", discute con tu compañera o compañero de grupo el trabajo realizado, verifica la solución que dieron a uno de los ejercicios planteados y consulta con tu docente las preguntas que aún tengas.

Completa los campos 1 y 2 del siguiente esquema.

Gráfico de anclaje. Sesión 2, guía 3, grado 5°



Grado 5° Guía 3

TIC

Sesión 3

Aprendizajes esperados

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

- Interpretar un diagrama de flujo para resolver problemas.
- Leer un código y reconocer si cumple o no requerimientos específicos previamente indicados.

Duración sugerida

15% 70% 15%



Material para la sesión

Material por grupo:

- Anexos 3.1 y 3.2.

Lo que sabemos, lo que debemos saber

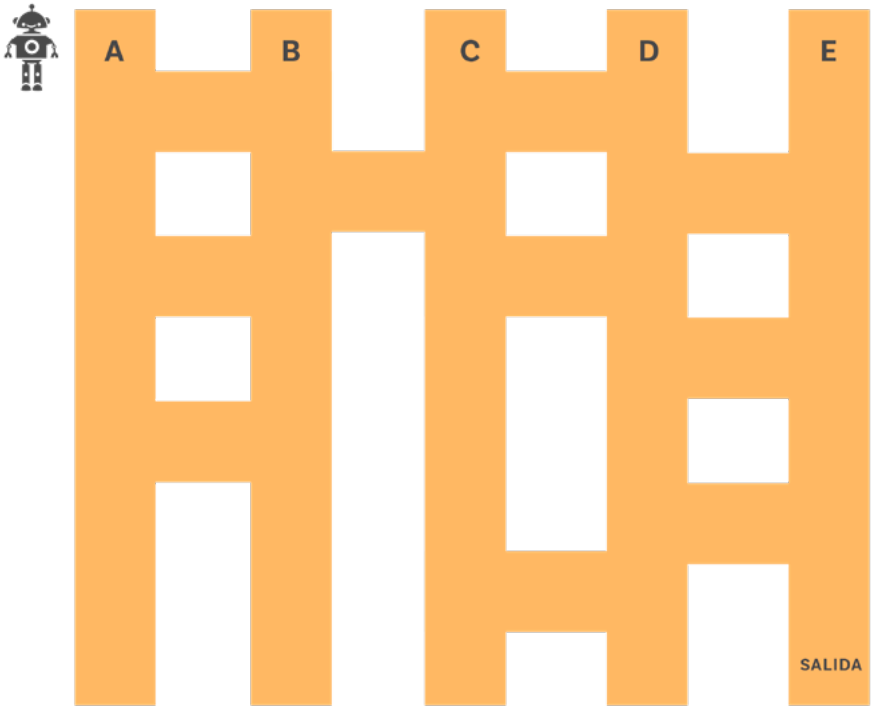
Esta sección corresponde al 15% de avance de la sesión

En la sesión pasada se mencionó que los condicionales permiten tomar decisiones dentro de los programas o algoritmos dependiendo de si se cumplen o no ciertas condiciones.

Ahora verás cómo estos condicionales permiten que dispositivos electrónicos como los robots puedan tomar decisiones de forma autónoma, es decir, sin ayuda externa, únicamente basándose en su programación.

El robot de la *Figura 1* ha sido programado para desplazarse a través de una red de tubos de alcantarillado. Se le va a enviar en una misión exploratoria para que llegue hasta el punto marcado como salida, que es donde se requiere ubicarlo para apoyar al personal que está encargado de realizar algunas reparaciones.

Figura 1. Robot y red de tubos

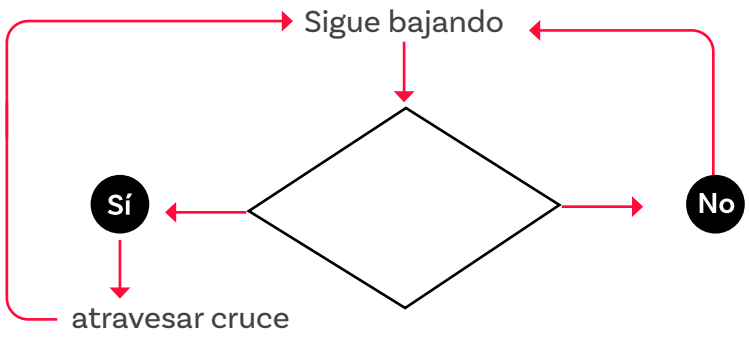


Este robot ha sido programado para ejecutar las siguientes instrucciones:

1. Entrar por un tubo.
2. Bajar por él hasta que encuentre un cruce a otro tubo, en cualquiera de los dos lados.
3. Si encuentra un cruce, debe atravesarlo.
4. Volver al paso 1.

Analiza la imagen y decide por cuál de los tubos debería ingresar el robot para llegar a la salida.

Antes de responder, completa el diagrama de flujo que refleje estas condiciones:



Ahora puedes probar el algoritmo para encontrar la respuesta correcta.

Probablemente notaste que, en ocasiones, seguir el algoritmo hizo que el robot se alejara de la salida y que, al comprobarlo, tú no quisieras atravesar algunos cruces. Sin embargo, debes recordar que, en la programación, los condicionales son los que deciden qué instrucciones se ejecutan. Por eso, siempre que se cumpla una condición, se va a ejecutar la instrucción.

Así como el del robot, existen algoritmos que ayudan a resolver acertijos o llegar a una meta. A continuación, podrás aplicar un algoritmo para salir de cualquier laberinto.

Anexos

Anexo 3.1



Anexo 3.2



Anexo 3.3



Manos a la obra

Desconectadas

Esta sección corresponde al 85% de avance de la sesión

Ahora vas a trabajar en grupo con tres personas más, según las indicaciones de tu docente.

Imaginen la siguiente situación.

Se han inscrito al grupo de espeleología[1] del municipio, en el que se entretienen explorando cavernas. En una de las salidas se encuentran con una cueva que no conocen y que parece tener muchos túneles. El grupo se pregunta si, al entrar, podrán salir sin tener un mapa. Ustedes saben que, usando la lógica de la computación, pueden lograrlo, y les cuentan de un algoritmo llamado "seguir la pared", que se muestra en el *Anexo 3.3*. El resto del grupo está escéptico, así que, para demostrarles que el algoritmo funciona con cualquier tipo de configuración, ustedes les piden que les den dos laberintos en papel, como los que se muestran en los *Anexos 3.1* y *3.2*.

Su misión, por tanto, es demostrar el algoritmo de seguir la pared, para llegar hasta la salida de los dos laberintos. Pueden replicar estos laberintos dibujándolos o marcándolos con cinta en el piso para que se puedan desplazar, o pueden usar una hoja con el dibujo de los laberintos y una ficha que represente su desplazamiento. Dentro de su grupo, asígne los roles de depurador(a), procesador(a), medidor(a) de complejidad y verificador(a). A continuación, se describe cada rol.

- **Depurador(a):** persona encargada de seguir el diagrama de flujo poniendo una ficha en la instrucción que se está ejecutando e indicándola en voz alta.

1. La espeleología es una ciencia que estudia las cuevas. Las personas que trabajan en esta ciencia son las y los espeleólogas(os), y se dedican a investigar las rocas, las plantas y los animales que encuentran dentro de las cuevas.

- **Procesador(a):** persona que se ubica a la entrada del laberinto, si está dibujado o hecho con cinta en el piso, o que coloca un objeto o ficha en la entrada del laberinto en papel. En el caso de usar un objeto, la ficha que use debe tener una cabeza o frente que le permita determinar cuál es su izquierda o derecha.

- **Medidor(a) de complejidad:** persona encargada de ir contando el número de pasos requeridos para salir del laberinto.

- **Verificador(a):** persona que comprueba que se siga la secuencia de instrucciones del algoritmo y, también, lleva cuenta de cuántos pasos se dan.

Cambien de roles cuando pasen al segundo laberinto.

Terminada la labor, comparen con otros grupos el número de pasos dados en los dos laberintos. Alternativamente, pueden buscar otro laberinto más complejo para probar el algoritmo y verificar si también pueden llegar hasta la salida siguiéndolo. Ya que lograron probar el algoritmo para salir de cualquier laberinto, recuerden que:

Los condicionales te permiten tomar decisiones. Son como las señales que ayudan a decidir qué camino seguir.



Trabajando con las compañeras y compañeros del grupo, piensen cómo se verían los diagramas de flujo de otras decisiones en la vida. Escojan una situación y dibujen su propio diagrama. Por ejemplo, podrían pensar en el diagrama de flujo para realizar alguna de las siguientes tareas:

- Elegir qué hacer después de clases.
- Elegir, entre varias opciones, un juguete que desean que les regalen.
- Elegir qué ropa ponerse un domingo.

Una vez tengan listos sus diagramas, sigan las indicaciones de su docente para compartirlos con los demás grupos y para poder ver sus propuestas.

Después, de forma individual, completa las siguientes frases sobre lo aprendido en esta sesión:

Algo que entiendo de los condicionales es:

_____

_____

_____

Una pregunta que tengo es:

_____

_____

_____

Me gustaría saber más acerca de:

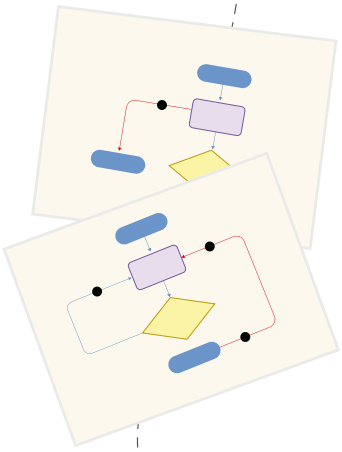
_____

_____

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes de la sesión. ¿Crees que lograste alcanzarlos?



1. ¿Puedes interpretar un diagrama de flujo para resolver problemas?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes leer algoritmos con tomas de decisión y reconocer si cumplen o no los requerimientos específicos para los que fueron creados?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", consulta con tu docente las inquietudes que tienes sobre el tema. Luego, observa el diagrama de flujo creado por alguno de los demás grupos. Léelo con atención y toma nota de lo que crees que debería pasar en cada punto. Cuando creas que tienes una idea clara de lo que debería hacer ese algoritmo, cuéntale lo que encontraste al grupo que lo creó. Aclara cualquier diferencia que haya entre tu comprensión del diagrama y lo que el grupo planteó que hiciera.



Sesión 4

Aprendizajes esperados

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

- Usar el sensor del acelerómetro como entrada para recuperar datos y enviarlos al microprocesador.
- Crear un programa que use más de dos condiciones.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Dispositivo con acceso a *MakeCode*.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En la sesión pasada, practicaste el uso y la creación de algoritmos para tomar decisiones, y para ello utilizaste los condicionales. Al finalizar la sesión, te retaste a crear un algoritmo para tomar decisiones en alguna situación de tu día a día.

Observa los diagramas de flujo en la *Figura 1* y responde:

¿En qué se parecen y en qué se diferencian?

¿Cuántos condicionales tiene cada algoritmo?

Figura 1. Algoritmo 1 y algoritmo 2

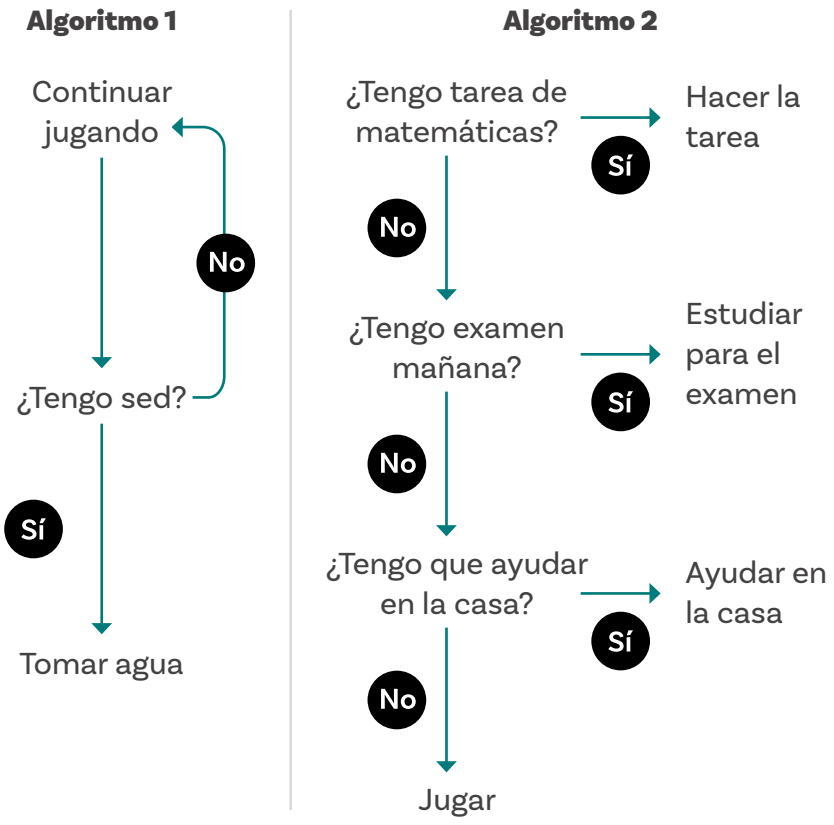


Figura 2. Bloque Para siempre



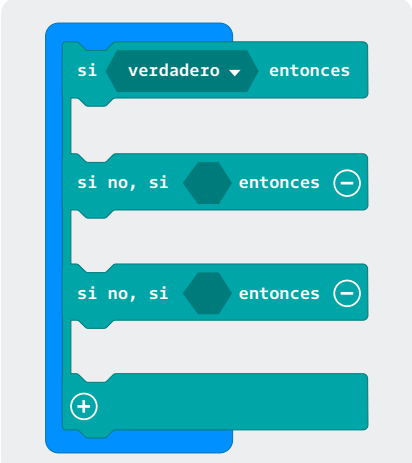
Figura 3. Bloque condicional "si"



Figura 4. Bloque condicional "si, si no"



Figura 5. Signo "+" para añadir más condiciones



Seguramente puedes ver que el algoritmo 1 tiene solo un condicional, mientras que el algoritmo 2 tiene tres condicionales, que se preguntan uno después del otro. El algoritmo 2 tiene en cuenta más opciones y es mucho más complejo que el número 1.

Asimismo, algunas aplicaciones necesitan programas simples y otras aplicaciones necesitan programas más complejos. Por ejemplo, una puerta automática como la que se encuentra en algunos supermercados o almacenes únicamente necesita abrirse cuando hay alguien cerca, esperar unos segundos y cerrarse. Pero un ascensor tiene muchas más condiciones para decidir si sube, baja, abre las puertas, cierra las puertas o hace sonar la alarma.

En la próxima actividad vas a trabajar creando un programa que requiere el uso de varios condicionales.

Recuerda que para que tu programa se repita una y otra vez, debes dejarlo en un bucle infinito con el bloque Para siempre (ver *Figura 2*).

Además, como has aprendido previamente, los bloques condicionales encajan perfectamente en los bloques de bucle. El hexágono que está al lado de la palabra clave "si" representa una variable booleana, por lo tanto, asume dos posibles valores: verdadero o falso. Los bloques que insertes en el bloque condicional si, solo se ejecutarán cuando el valor del hexágono sea verdadero (ver *Figura 3*).

El bloque condicional si, si no tiene un espacio adicional para agregar bloques que se ejecutarán cuando el valor del hexágono sea falso. Puedes convertir el bloque "si" en uno "si, si no" presionando el signo "+" de la parte inferior (ver *Figura 4*).

Si presionas otra vez el signo "+" puedes añadir más condiciones , o puedes eliminarlas presionando el signo "-" (ver *Figura 5*).

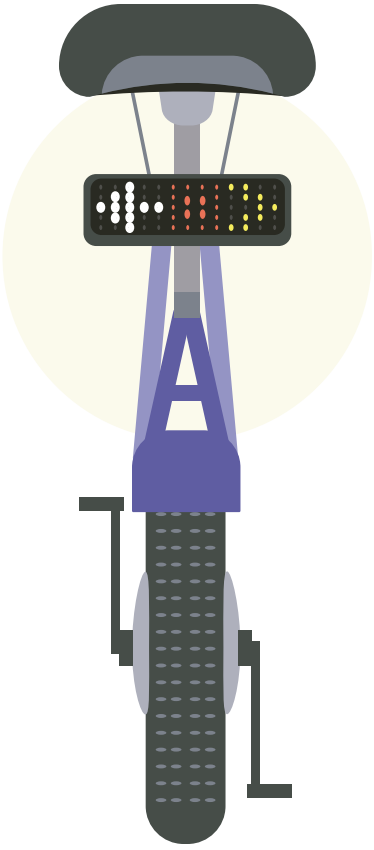
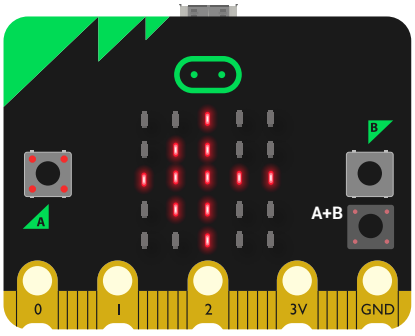


Figura 6. Luz direccional a la izquierda



Manos a la obra

Conectadas

Esta sección corresponde al 85% de avance de la sesión

Vas a trabajar con otra persona, siguiendo las instrucciones de tu docente.

Con tu grupo van a programar el sistema de direccionales para una bicicleta. Este dispositivo puede ayudar a las personas que se desplazan en bicicleta a transitar de forma más segura por las vías permitidas.

El dispositivo debe mostrar luces informativas mientras se monta en bicicleta. Si se creara en una versión final, los botones de la *micro:bit* podrían extenderse para presionarlos desde los manubrios, donde se colocan las manos. Sin embargo, con fines ilustrativos y a modo de prueba de concepto de tu diseño, programarán los botones A y B que ya conocen.

El dispositivo deberá funcionar de la siguiente manera:

1. Cuando se presione el botón A, la *micro:bit* deberá mostrar una luz direccional a la izquierda parpadeando.
2. Cuando se presione el botón B, la *micro:bit* deberá mostrar una luz direccional a la derecha parpadeando.
3. Cuando se presionen los dos botones, la *micro:bit* deberá mostrar un indicativo de frenar para alertar a quien va detrás. Elijan un ícono que les parezca apropiado para transmitir ese mensaje.

El código en la *Figura 7* puede inspirarles para empezar su propio código.

Es hora de programar y simular en el editor *MakeCode* el código para verificar su funcionamiento.

Figura 7. Código de ejemplo



Cuando tu código incluye la condición de A+B, en el simulador aparece un tercer botón para probar tu código.



1. ¿A qué parte del funcionamiento corresponde el programa anterior?
2. Al programarlo en el editor *MakeCode*, ¿qué sucede si retiran el bloque "borrar la pantalla"?
3. Ahora deben programar el resto del funcionamiento. En este mismo bloque "para siempre", agreguen las condiciones faltantes.

¿Lograron solucionar el reto planteado?

Trabajando con otras entradas

En sesiones anteriores trabajaron con el sensor de temperatura y el de luz. Recuerden que estos sensores son otras variables de entrada. ¡Pero la *micro:bit* posee más sensores! Para el siguiente reto usarán el sensor llamado **acelerómetro**. Este sensor, que se muestra en la *Figura 8*, mide de cierta forma el movimiento del dispositivo.

Figura 8. Acelerómetro



Imaginen ahora que tienen una botella llena de agua y que al taparla queda una burbuja de aire atrapada en su interior.

A medida que cambia de posición la botella, la burbuja se desplaza para quedar siempre lo más arriba posible. La burbuja se mueve tan rápido como muevan la botella. Así, el acelerómetro también puede saber en qué posición se encuentra la *micro:bit*: con el logotipo arriba, abajo, inclinado, etc.

El programa que crearon en la sección anterior puede ser muy útil cuando monten en bicicleta, para indicar a quienes estén a su alrededor si van a girar a la izquierda o a la derecha, si están en movimiento o si van a parar. Haciendo algunos cambios pueden hacer el programa aún mejor.

¿Han notado que al montar en bicicleta se inclinan ligeramente hacia el mismo lado hacia el que están girando?

¿Cómo creen que pueden usar este fenómeno para mejorar su programa utilizando los gestos de la micro:bit?

Además de medir si hay cambios en el movimiento, el acelerómetro puede indicar si la *micro:bit* está completamente horizontal o inclinada hacia la izquierda o hacia la derecha. Si fijan la *micro:bit* a su cuerpo no necesitarían presionar los botones A o B para indicar un giro, la *micro:bit* podría usar su acelerómetro para determinar si están girando y en qué dirección.

Usen el bloque que se encuentra en el menú “Entrada”, ver *Figura 8,* para elegir las opciones más adecuadas y mejorar su programa.

Glosario

Acelerómetro: es un sensor que permite medir la aceleración y el movimiento de un objeto. En la *micro:bit* este sensor puede indicar si se está moviendo más rápido, más lento o si está cambiando de dirección. Los teléfonos móviles también cuentan con un acelerómetro, que les indica si los estás moviendo o girando.

Figura 9. Flechas que indican desplazamiento hacia adelante



Para ir más lejos

Pueden usar el acelerómetro para hacerse más visibles mientras estén montando bicicleta.

Teniendo en cuenta que el dispositivo se estará agitando a medida que se mueven, podrían mostrar flechas que indiquen que se están desplazando hacia adelante, como se muestra en la *Figura 9*.

Para ilustrar un desplazamiento hacia adelante

1. El bucle presentado en el diagrama de la izquierda es un bloque general, como el bucle “para siempre”. Dentro de este bucle hay dos bloques que se repiten uno tras otro mientras se cumpla la condición, es decir, estar agitado.

2. Antes de implementar el diagrama mencionado, intenten predecir lo que ocurrirá al ejecutarlo. En este caso, la entrada “agitado” del acelerómetro es una variable booleana que se encuentra en la sección de entradas.

3. Al utilizar el sensor acelerómetro, el simulador presenta un botón que dice “shake” (agitar) para representar que la tarjeta está siendo agitada.

4. ¿Ven la flecha desplazarse? ¿Podrían mejorar este desplazamiento? Como aprendieron previamente, una animación requiere al menos dos imágenes que se muestran una tras otra y dan la sensación de movimiento, pero pueden agregar más imágenes intermedias para hacer el movimiento más fluido.

5. Complementen el sistema de luces agregando animaciones para todos los indicadores.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes de la sesión. ¿Crees que lograste alcanzarlos?

1. ¿Puedes usar el sensor del acelerómetro como entrada para recuperar datos y enviarlos al microprocesador?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes crear un programa que use más de dos condiciones?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

> **Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", revisa el trabajo realizado y consulta con tu docente las preguntas que aún te quedan sobre el tema.**

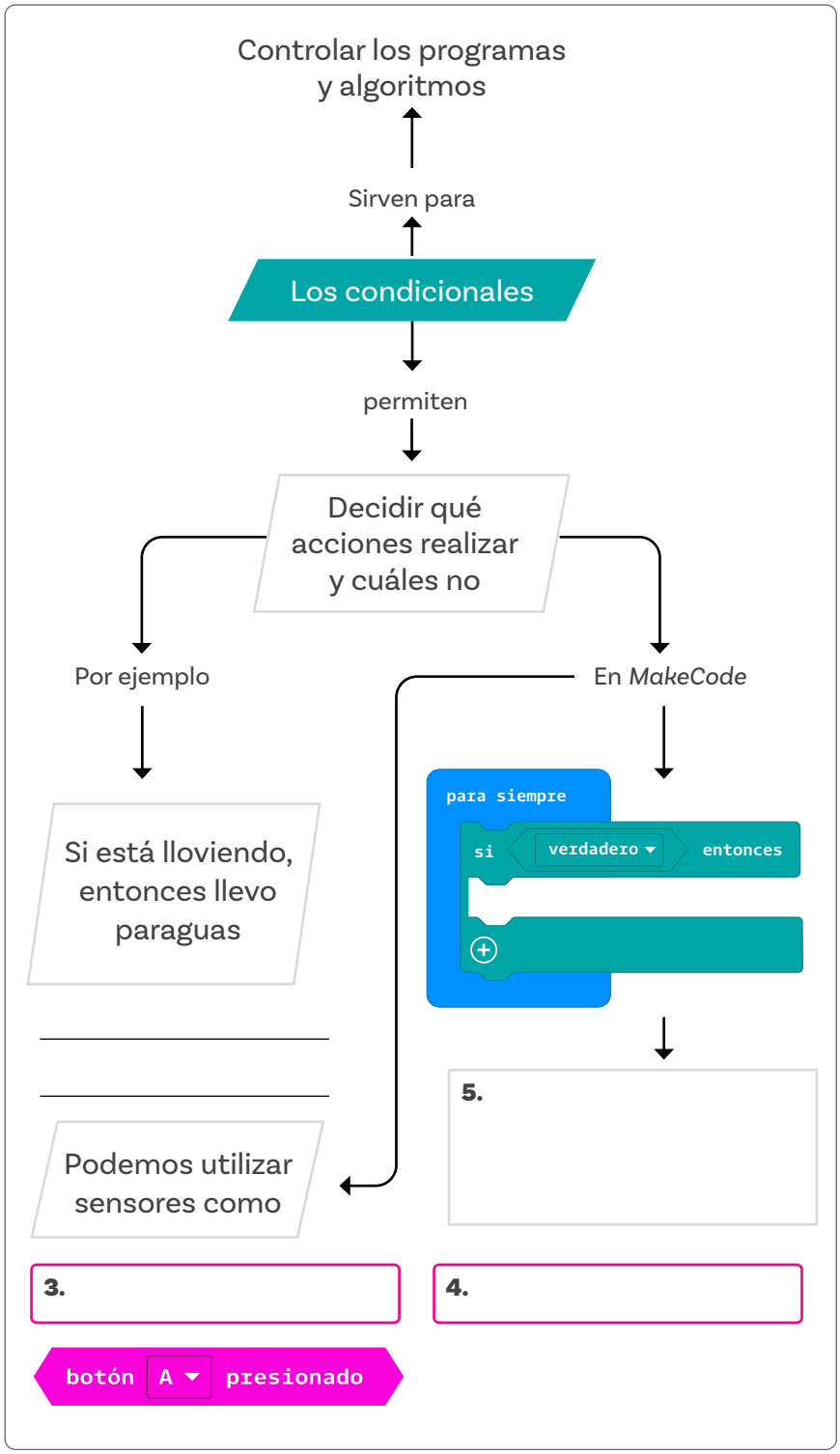
Actividad opcional para seguir consolidando aprendizajes

Con las actividades realizadas en esta sesión continuaste fortaleciendo lo aprendido sobre la programación de condicionales en *MakeCode*. Además, trabajaste con otro sensor de la *micro:bit*, que te permite determinar la dirección del movimiento de la tarjeta.

Imagina que le vas a contar lo que hiciste en la clase de hoy a una prima o primo menor que tú. ¿Qué le dirías? Crea un dibujo que represente el trabajo realizado y lo que aprendiste o recordaste gracias a él.

Continuemos con nuestro gráfico de anclaje; completa los campos 3, 4 y 5.

Gráfico de anclaje. Sesión 4, guía 3, grado 5°



Grado 5° Guía 3

TIC

Sesión 5

Aprendizajes esperados

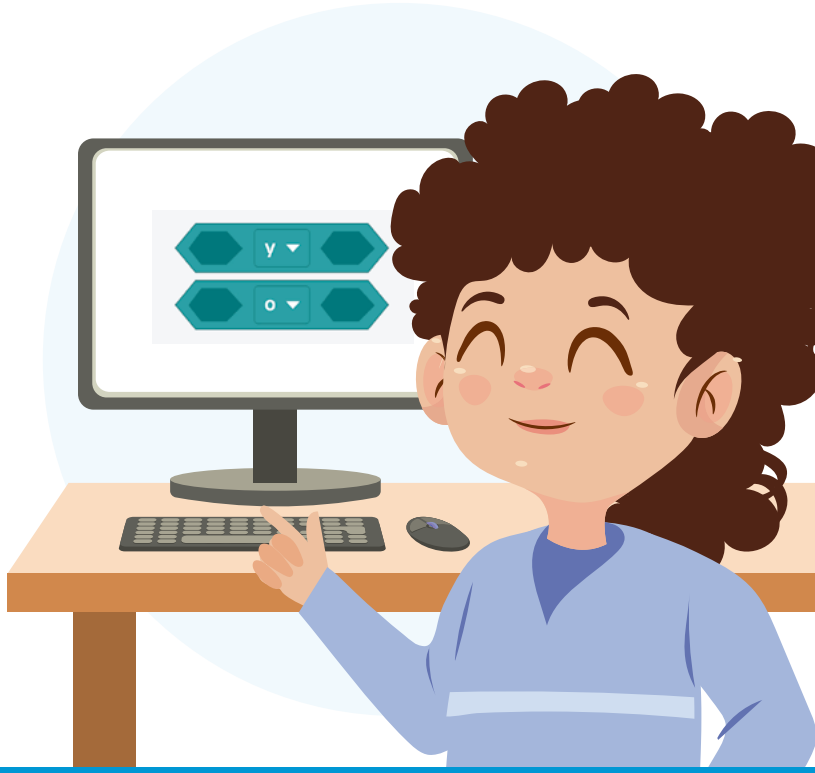
Con las actividades de esta sesión se espera que puedas avanzar en:

- Utilizar los operadores lógicos “Y” y “O” para evaluar más de dos condiciones.
- Realizar lecturas de datos a través de diferentes sensores de la *micro:bit* y utilizarlos en comparaciones asociadas a tomas de decisiones.
- Crear un programa que use condicionales compuestos.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Dispositivo con acceso a *MakeCode*.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

A veces se necesita que varias condiciones se cumplan al mismo tiempo para cumplir algún objetivo.

Por ejemplo, si se quiere preparar arroz con leche, hay algunos ingredientes que son obligatorios porque se requieren todos:

Si quieres arroz con leche, debes tener: arroz, leche **Y** una olla para cocinar.

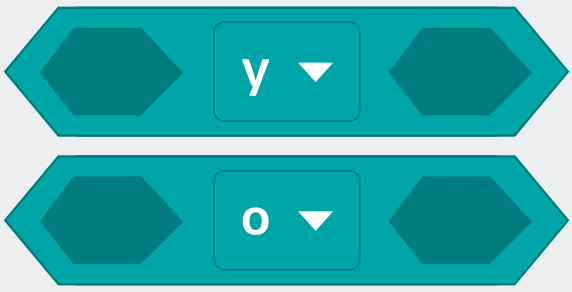
Pero hay otros ingredientes que son opcionales:

Dependiendo de lo que te guste, puedes agregar: panela **O** queso **O** canela.

Las letras "Y" y "O" son **operadores lógicos**, y, junto a los condicionales, te ayudan a controlar los algoritmos.

En *MakeCode* se ven como en la *Figura 1*.

Figura 1. Operadores lógicos en *MakeCode*



En las siguientes actividades practicarás cómo usarlos.

Glosario

- **Operadores lógicos:** son palabras o símbolos, utilizados en programación o lógica matemática para combinar, comparar o cambiar variables. Los operadores lógicos más conocidos son “Y” y “O”.
- **Y:** operador lógico que solo da resultados verdaderos si se cumplen todas las condiciones que une.
- **O:** operador lógico que da resultados verdaderos si se cumple al menos una de las condiciones que une.

Manos a la obra
Desconectadas



Esta sección corresponde al 85% de avance de la sesión

Trabaja con una compañera o compañero para realizar las siguientes actividades.

Actividad 1

Lean cada adivinanza cuidadosamente y traten de encontrar la solución. Cada adivinanza contiene un condicional que les dará una pista sobre la respuesta. ¡Diviértanse resolviendo las adivinanzas y descubriendo qué se esconde detrás de cada una!

Si tiene cuatro patas y maúlla en la noche, ¿qué animal puede ser?

Respuesta: _____

Si un número es múltiplo de 5 y menor a 13, ¿qué número es?

Respuesta: _____

Si tiene hojas verdes y da sombra en verano, ¿qué objeto puede ser?

Respuesta: _____

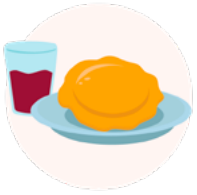
Si tiene ruedas y te lleva de un lugar a otro, ¿qué puede ser?

Respuesta: _____

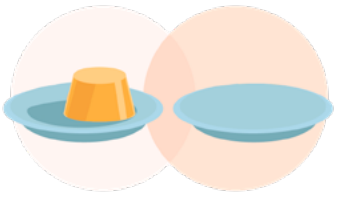
Actividad 2

Imaginen que están trabajando en un restaurante. Como saben que la chef está muy ocupada, tienen que revisar muy bien cada pedido. En el lado izquierdo están las órdenes, y en el lado derecho, la imagen de lo que entregó la chef. Marquen con un ✓ si el pedido es correcto o una ✖ si es incorrecto.

Pedido #1:
Una arepa de huevo Y un vaso de jugo de corozo



Pedido #2:
Una porción de natillas
O una empanada



Pedido #3:
Un pan de bono, un pan de yuca, un café con leche Y una porción de queso



Pedido #4:
Una ensalada O una limonada



Pedido #5:
Un tamal con arepa Y chocolate



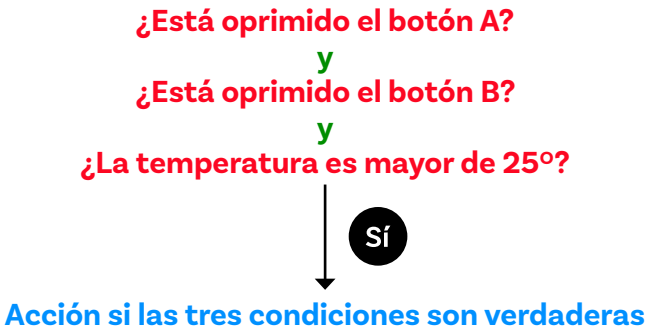
Figura 2. Diagrama de flujo



Manos a la obra

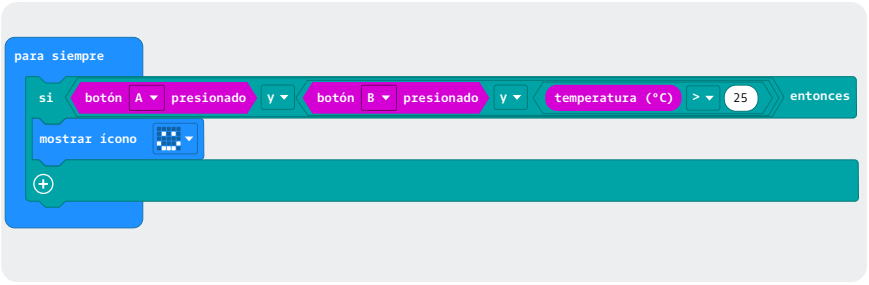
Conectadas

Algunos algoritmos tienen dos o tres condiciones seguidas, como en el diagrama de la *Figura 2*. En estos casos se pueden reemplazar varias condiciones por una sola verificación, que incluya las dos o tres condiciones. A esto se le conoce como **condicional compuesto**:



Lo mismo puede ser expresado por un solo bloque condicional con varias condiciones, como el que se muestra en la *Figura 3*. Para que se visualice la cara feliz se requiere que las tres condiciones sean verdaderas. Si alguna es falsa, no debe mostrarse la cara feliz:

Figura 3. Bloque condicional con varias condiciones



¡Es hora de programar! Trabaja con una compañera o compañero siguiendo las indicaciones de tu docente. Codifiquen el programa anterior.

Figura 4. Maceta inteligente



¿Les funciona como esperaban?

Ahora modifiquen este programa cambiando las Y por O. ¿Qué pasa?

En este caso, la cara feliz debe mostrarse cuando una o varias de las condiciones sean verdaderas, aunque no se cumplan las tres al tiempo.

Ahora su reto será programar la *micro:bit* para que simule una maceta inteligente de la siguiente manera:

1. Cuando la maceta se agite o se caiga, debe mostrar una cara asustada.
2. Cuando esté oscuro (luz igual o inferior a 100) y la temperatura sea menor a 28 grados, debe mostrar una cara dormida.
3. Cuando haya suficiente luz (mayor a 100) y la temperatura sea mayor o igual a 28 grados, debe mostrar una cara feliz.
4. Cuando ninguna de las condiciones anteriores se cumpla, debe mostrar un ícono elegido por ustedes.

Glosario

Condicional compuesto: es un condicional que cuenta con 2 o más condiciones conectadas con operadores lógicos. Se deben evaluar todas las condiciones que forman la expresión para determinar si el resultado es verdadero o falso.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes de la sesión. ¿Crees que lograste alcanzarlos?

1. ¿Puedes reconocer los operadores lógicos "Y", "O" para evaluar más de dos condiciones?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes crear un programa que use condicionales compuestos?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

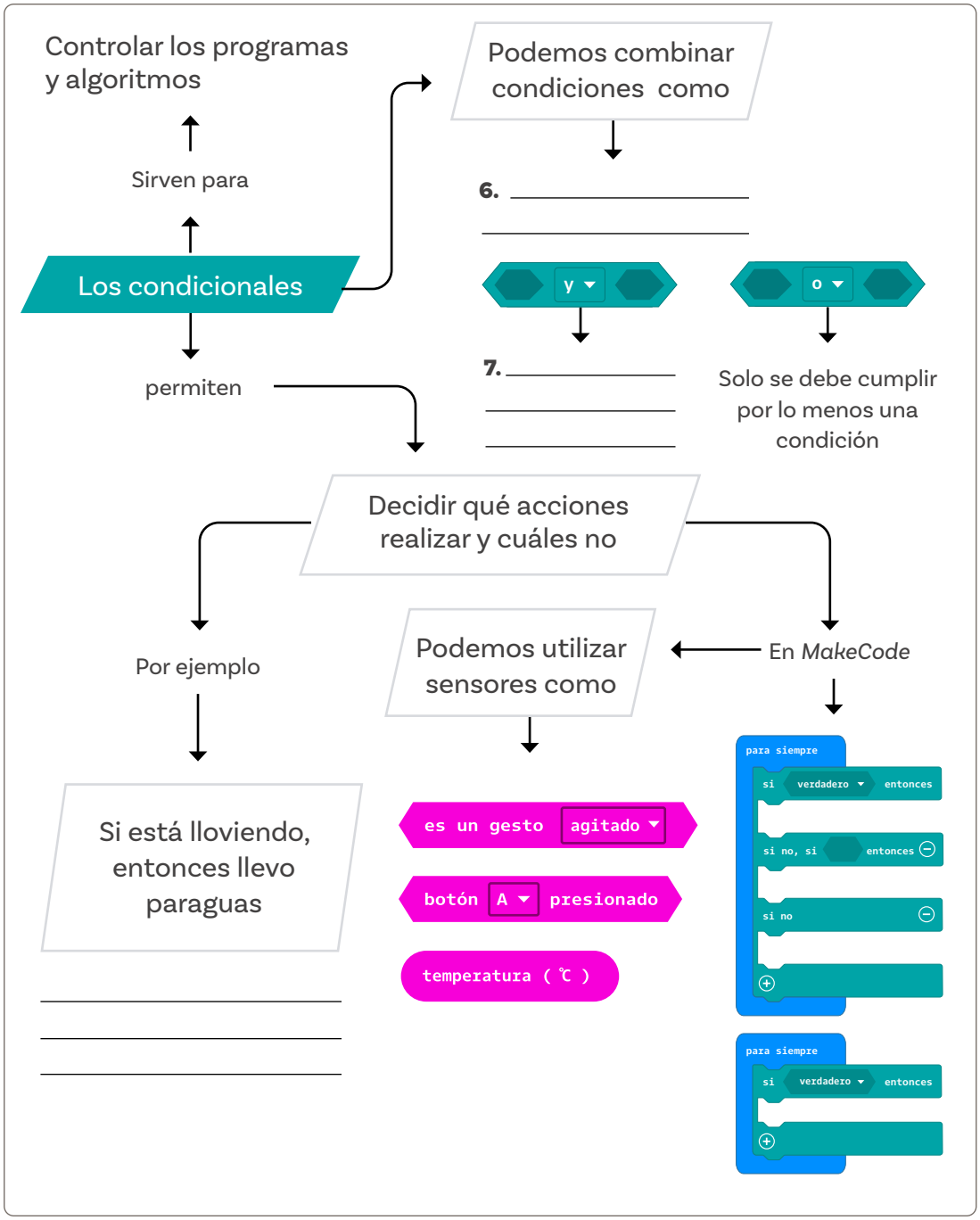
3. ¿Puedes realizar lecturas de datos a través de diferentes sensores de la *micro:bit* y utilizarlos en comparaciones asociadas a tomas de decisiones?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", revisa nuevamente las actividades desarrolladas en la guía. Anota las preguntas que todavía tienes sobre el tema. Compártelas con tu docente. Una vez las hayas aclarado, intenta resolver por ti misma(o) el reto de la matera inteligente, pero cambiando las condiciones según te indique tu docente.

Completa el gráfico de anclaje y agrega todas las explicaciones y ejemplos que puedas, para ayudarte a recordar.

Gráfico de anclaje. Sesión 5, guía 3, grado 5°



Actividad opcional para seguir consolidando aprendizajes

Trabaja en grupo con dos o tres personas, según indique tu docente, y elijan un nombre creativo para su equipo. Participarán en una competencia de velocidad basada en lo aprendido en esta guía.

Tu docente hará las mismas preguntas a todos los equipos:

A. ¿Para qué sirven los condicionales?

B. ¿Cómo se llama a letras Y u O que se usan para conectar condiciones?

C. ¿En qué sección de la caja de herramientas de *MakeCode* se encuentran los bloques de condicionales?

D. ¿Cómo se llama el bloque que permite usar el sensor de luz de la *micro:bit*?

E. ¿Qué hace el acelerómetro?

F. ¿Qué hace el código en la *Figura 5*?

Figura 5. Código

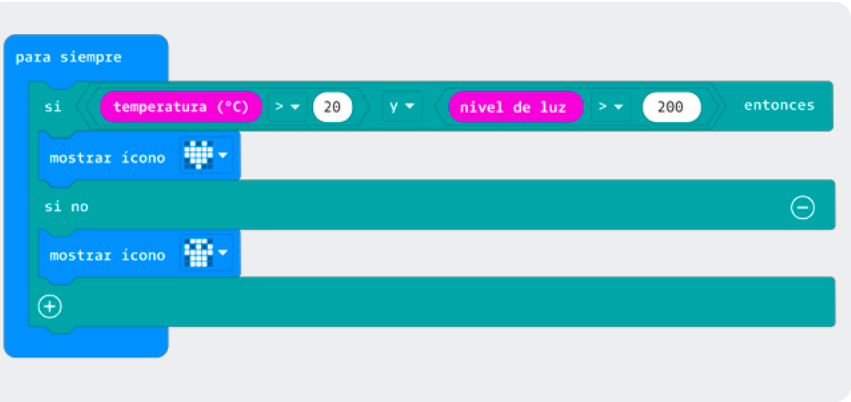


G. Den un ejemplo de condicional compuesto.

H. Den un ejemplo de condicional Si... entonces, Si no...

I. ¿Qué mostrará la *micro:bit* en pantalla si está programada con el código en la *Figura 6* y hay 15 grados de temperatura y 230 de nivel de luz?

Figura 6. Código



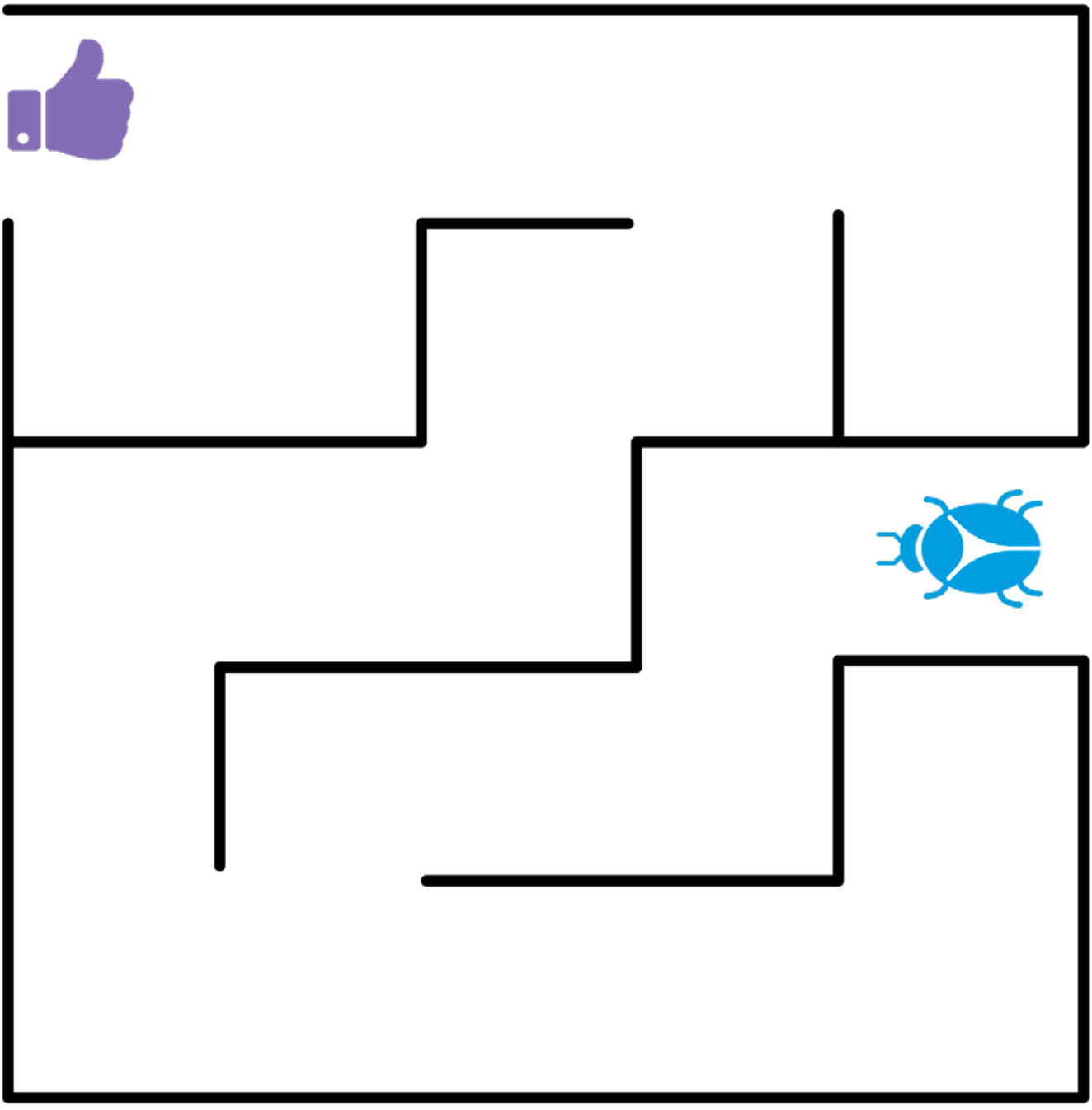
Las reglas del juego son las siguientes:

- Tu docente leerá cada pregunta en voz alta.
- El primer equipo en responder correctamente gana un punto.
- Lleven un registro de sus puntos durante el juego.

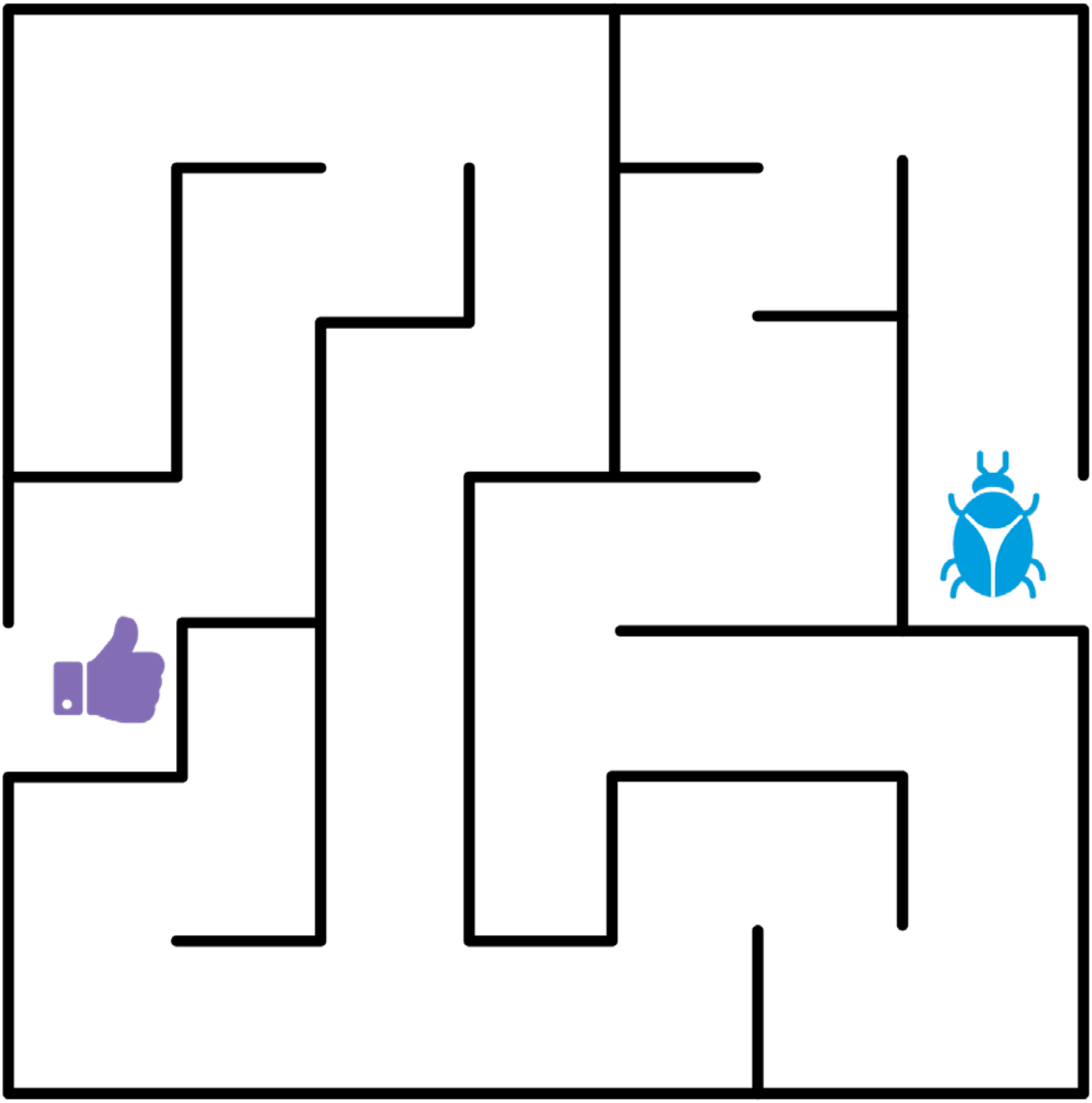
Al final de la competencia, el equipo con más puntos será el ganador de la sesión.



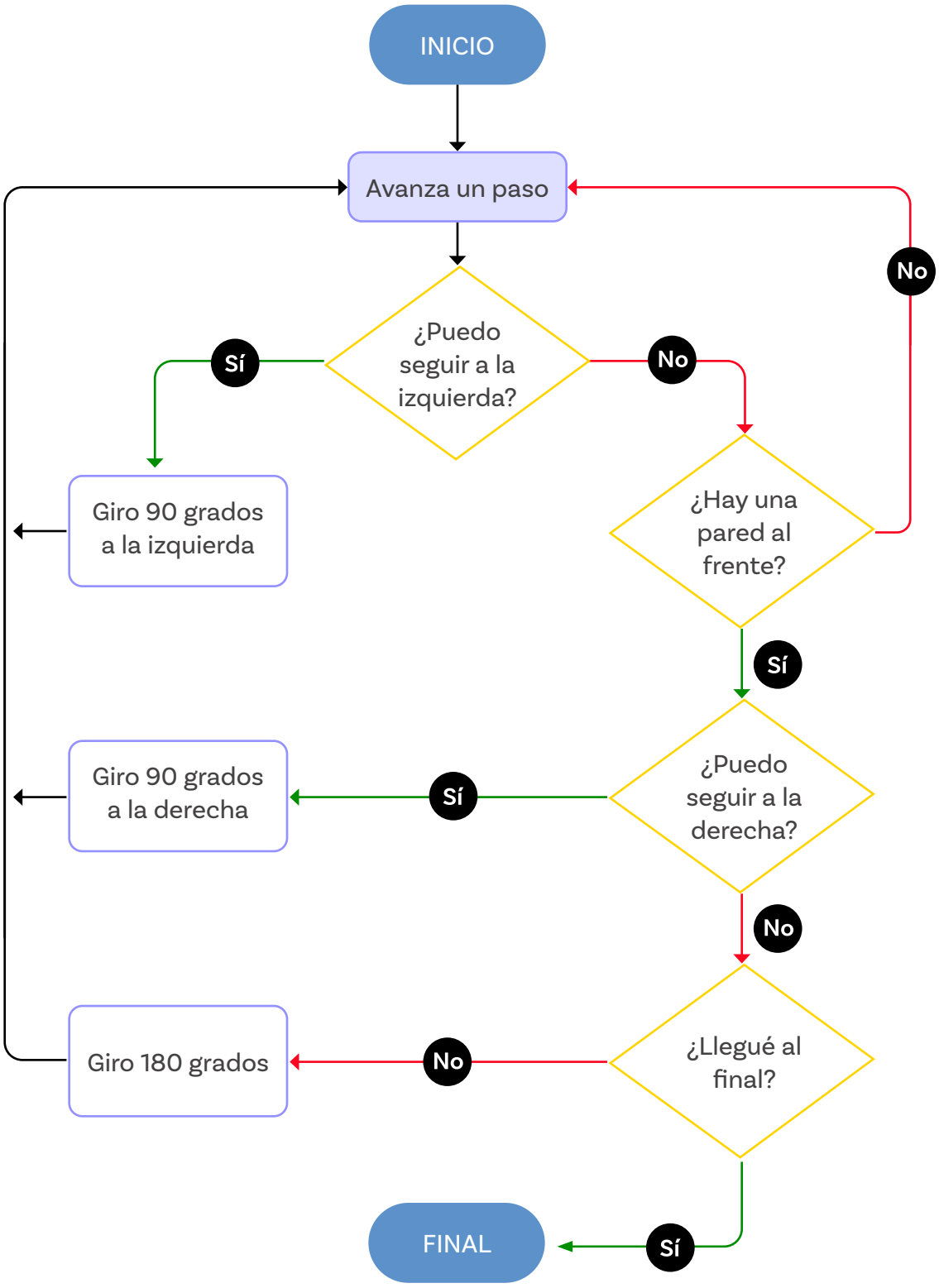
Anexo 3.1 Seguir la pared - 1



Anexo 3.2 Seguir la pared - 2



Anexo 3.3 Algoritmo seguir la pared



TIC

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}