Luces, códigos y mensajes visuales: introducción a MakeCode

Grado 5° Guía 1

TIC



Estudiantes

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

Luces, códigos y mensajes visuales: introducción a MakeCode

Grado 5° Guía 1

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: 978-628-7934-61-0

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.

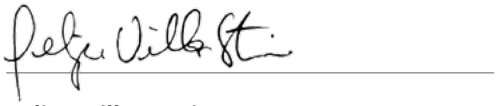
Encontrarán también en estas páginas orientaciones y sugerencias sobre cómo adaptar las actividades a diferentes contextos y necesidades. Por ejemplo, cómo desarrollar habilidades de pensamiento computacional en instituciones sin conectividad o incluso sin electricidad; cómo fomentar la equidad de género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

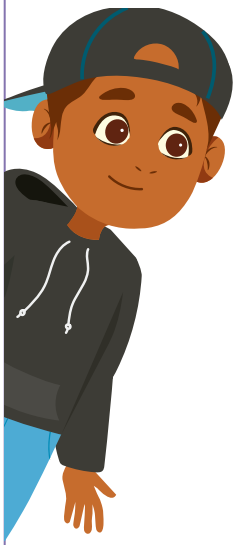
Creación

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.

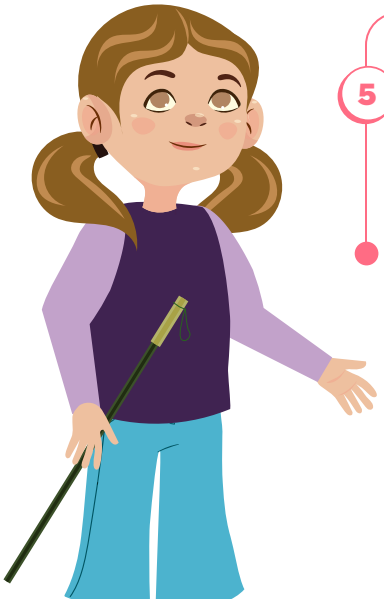
Mayo a diciembre 2024
Se incorporan a las guías los ajustes provenientes de las mesas de validación.

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 1

TIC

Identificación de los aprendizajes

- Lógica, programación y depuración
- Algoritmos, patrones, abstracción y descomposición

Aprendizajes de la guía

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

- Descomponer las tareas en otras más pequeñas y manejables para facilitar su solución.
- Anticipar los resultados de programas o algoritmos sencillos y reconocer cuando una salida o resultado no es el esperado.
- Describir los pasos y las decisiones que se tomaron durante el proceso de diseño y desarrollo de un programa.

Resumen de la guía

Esta guía propone 5 sesiones de trabajo con las que aprenderás a crear, por medio de programación por bloques, mensajes visuales con texto, números e imágenes que se pueden reproducir en un minicomputador portable llamado *micro:bit*.

Resumen de las sesiones

Sesión 1

Se realiza un pequeño programa, con un lenguaje de flechas, para simular el movimiento de un robot de apoyo en actividades de cuidado medioambiental.

Sesión 2

Se presenta la tarjeta *micro:bit* y se reconoce el entorno de uno de los editores de código en bloques para programarla, *MakeCode*. Con este fin, se crea un programa que incorpora tanto íconos prediseñados como originales.

Sesión 3

Se programan animaciones sencillas para acompañar y representar de forma gráfica ejercicios narrativos.

Aprendizajes de la guía

- Diseñar, escribir y depurar programas que permitan mostrar mensajes visuales en la pantalla de un microprocesador como la *micro:bit*.
- Usar entradas (botones) y salidas (pantalla) de un microprocesador como la *micro:bit* para recuperar o introducir información.

Sesión 4

Se continúa profundizando las opciones de visualización de la *micro:bit*, mediante la creación de mensajes de texto programados.

Sesión 5

Se aprenden a manejar los botones de la *micro:bit* al tiempo que se comienza a trabajar condicionales.

Conexión con otras áreas

A continuación, se presenta la conexión con otras áreas:

Matemáticas

- En la sesión 1 se trabaja en desplazamiento en un plano mientras en la sesión 5 se trabaja con condicionales lógicos, aprendizajes que se conectan con el plano cartesiano y con lógica matemática.

Nota

Esta guía es una adaptación de la ficha “Luces y Códigos”, 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 1

TIC

Sesión 1

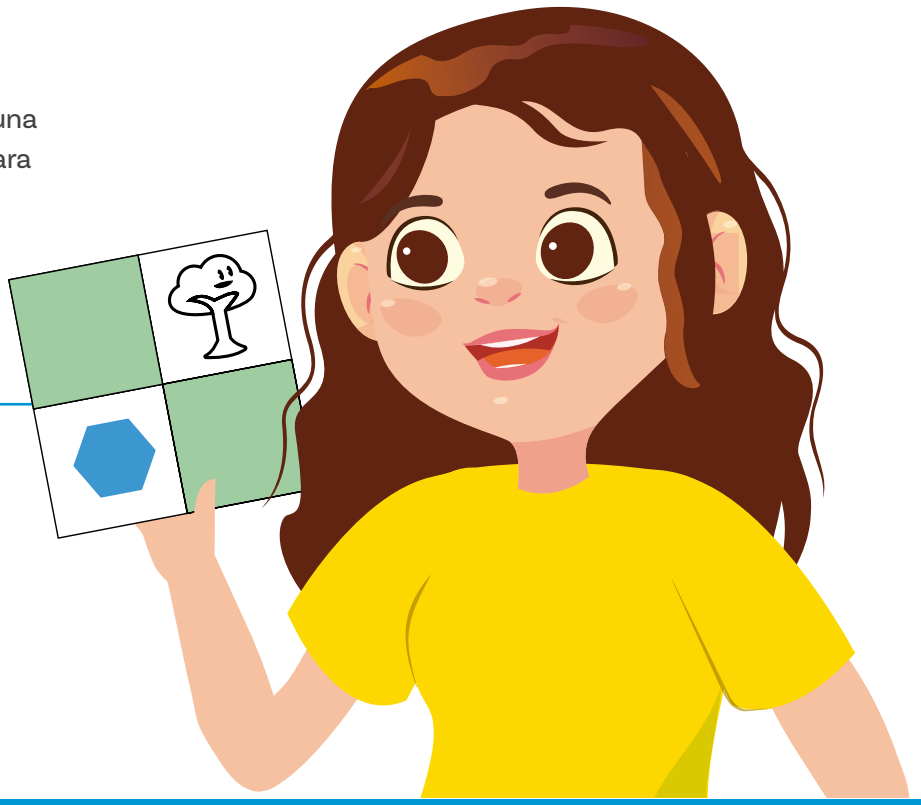
Aprendizajes esperados

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

- Identificar en una actividad de programación desconectada los conceptos de algoritmo, programa, procesador y programador(a).
- Proponer, ejecutar y verificar una secuencia de instrucciones para realizar una tarea.

Duración sugerida

15% 70% 15%



Material para la sesión

Material por grupo:

- Anexos 1.1, 1.2, 1.3.
- 12 fichas.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Piensa por un momento en la palabra computador. Es posible que lo primero que se te venga a la mente sea una pantalla, la de un computador portátil o la de un computador de escritorio. ¿Pensaste quizás en algo diferente, como un celular? Piensa en otros ejemplos de aparatos y dispositivos que pudieran asociarse a esa palabra. ¿Relacionaste la palabra computador con un reloj digital, un horno microondas o una lavadora? Si así lo hiciste, seguro tienes claro que un computador es una máquina electrónica con la capacidad de realizar una o varias tareas que se le indiquen.

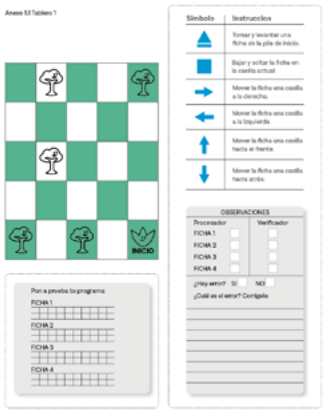


Ahora piensa por un momento en una lavadora. ¿Cómo determina una lavadora el momento de iniciar el ciclo de lavado? ¿Cuándo dejar caer el detergente? Cuando oprimes el ciclo de lavado suave, por ejemplo, la lavadora determina la cantidad de agua, su temperatura y cuántas veces gira su tambor, entre otras acciones. Pero ¿cómo logra esto? Una posible respuesta es que la lavadora necesita que generemos unas instrucciones para que ejecute las acciones correspondientes de manera lógica y secuencial. Como quizás recuerdas, estas instrucciones se llaman **algoritmos**.

Las lavadoras modernas incluyen **procesadores** que ejecutan algoritmos, estos deben estar escritos en un lenguaje específico llamado **programa**, el cual es desarrollado por una persona, conocida como **programadora o programador**. En el ejemplo de la lavadora, al elegir el ciclo de lavado suave, se ejecuta un programa con las instrucciones para que empiece a lavar, escurrir y centrifugar.

Anexo

Anexo 1.1



Glosario

- **Algoritmo:** conjunto de instrucciones secuenciales con el fin de describir los pasos en un proceso.
- **Programa:** conjunto de algoritmos escritos en un lenguaje específico para ser interpretados por un procesador.
- **Programador(a):** persona que diseña programas para que un procesador ejecute.
- **Procesador:** dispositivo electrónico que entiende programas y los ejecuta automáticamente.
- **Verificador(a):** persona encargada de comprobar que un programa cumpla su objetivo.

Manos a la obra

Desconectadas

Esta sección corresponde al 85% de avance de la sesión

Para recordar mejor estos conceptos, vas a trabajar en grupos de 2 a 3 personas, según las instrucciones de tu docente.

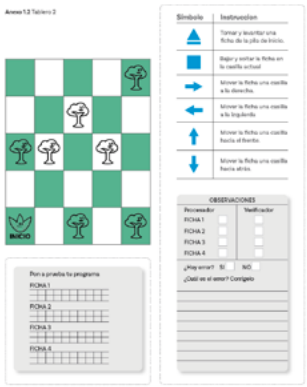
La misión de tu equipo será "simular" la programación y movimiento de otro dispositivo que tiene un computador dentro, un robot que puede evitar obstáculos dentro de un bosque y ubicar cámaras para estudiar las especies animales que lo habitan.

En esta simulación, las cuadrículas en los anexos (*Anexos 1.1, 1.2 y 1.3*) corresponden a segmentos del "bosque" que se van a estudiar y las fichas[1] representan las "cámaras" que se van a ubicar allí.

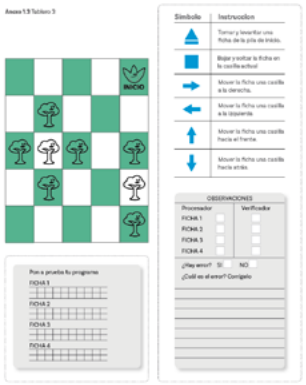
1. Como fichas puedes usar monedas, semillas, tapas, fichas de parqués u otro elemento semejante.

Anexos

Anexo 1. 2



Anexo 1. 3



Cada persona del grupo necesitará 1 copia diferente de uno de los anexos y 4 fichas. Se irán rotando roles en la medida que avanzan con la actividad, así:

1. Cliente: este es el primer rol que debe asumir cada persona del equipo. Decide dónde deben ubicarse las 4 "cámaras" en su mapa del "bosque". Marca las posiciones deseadas dibujando un símbolo como este ⬣ en las casillas libres (sin plantas) que elija en el mapa. Una vez hayas marcado las 4 ubicaciones esperadas, pásale tu mapa a otra persona del grupo, quien ahora será la encargada de programar los movimientos del robot.

2. Programador(a): este segundo rol inicia cuando todas y todos los clientes del grupo hayan intercambiado mapas. Tras recibir de otra persona el mapa con las ubicaciones deseadas para las cámaras, escribe el programa para el robot, teniendo en cuenta que:

   c. Todos los recorridos del robot comienzan en su punto de recarga (la casilla "Inicio") y finalizan en uno de los puntos elegidos para ubicar una cámara.

   d. El robot no puede moverse por encima de las plantas ni de fichas que ya estén en el tablero.

   e. El programa se escribe utilizando los símbolos de las instrucciones que se muestran en la Figura 1, dentro del espacio "Pon a prueba tu programa" del anexo. El Ejemplo 1 muestra cómo podría verse un programa como el que debes crear.

   Como observas en el ejemplo, el programa consiste en una secuencia de símbolos que dirá al procesador lo que debe hacer. Al programar, decides el orden en que se deben ubicar las fichas y determinas los movimientos que realizará el robot. Cuando termines, pasa el anexo a otra persona de tu equipo (no puedes pasarlo todavía a tu "cliente"). Es momento de que tomes el rol de procesador(a).

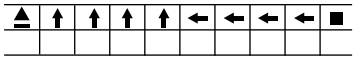
Figura 1. Lenguaje de flechas para programar los desplazamientos

Símbolo	Instrucción
⏏	Tomar y levantar una ficha de la pila de inicio.
■	Bajar y soltar la ficha en la casilla actual.
→	Mover la ficha una casilla a la derecha.
←	Mover la ficha una casilla a la izquierda.
↑	Mover la ficha una casilla hacia el frente.
↓	Mover la ficha una casilla hacia atrás.

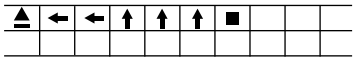
Ejemplo 1. Posibles programas para mover las fichas

Pon a prueba tu programa

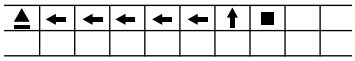
FICHA 1



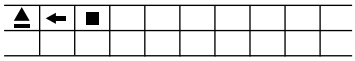
FICHA 2



FICHA 3



FICHA 4



Ejemplo 2. Marcado del éxito de los programas después de ejecutarlos

OBSERVACIONES		
Procesador		Verificador
FICHA 1	✓	☐
FICHA 2	✓	☐
FICHA 3	X	☐
FICHA 4	✓	☐

3. Procesador(a): en este rol harás las veces del computador interno del robot. Por tanto, debes leer el programa que recibiste y ejecutar las instrucciones que se indican allí para mover y ubicar cada una de las fichas. A medida que leas y ejecutes un programa, marca si se logró posicionar la ficha en la posición deseada ✓ o no X. Registra estas observaciones en la casilla correspondiente al procesador, como se muestra en el *Ejemplo 2*. Una vez hayas terminado de ejecutar los 4 programas, devuélvele el mapa a quien hizo de "cliente" original, sin dejar caer las fichas que están allí ubicadas. Ahora tú y las demás personas del grupo tendrán la tarea de verificar los programas.

4. Verificador(a): sigue uno a uno los movimientos indicados en los programas y comprueba que las fichas hayan quedado en el lugar indicado en cada caso. Registra los resultados en la casilla de verificación, al lado de los ya indicados por la persona que hizo las veces de procesador. Luego, indica si alguno de los programas generó errores y si es así, corrige. Observa el *Ejemplo 3*.

Ejemplo 3. Marcado del éxito de los programas después de verificarlos y depurarlos

OBSERVACIONES		
	Procesador	Verificador
FICHA 1	☑	☑
FICHA 2	☑	☑
FICHA 3	☒	☒
FICHA 4	☑	☑

¿Hay error? SI ☑ NO ☐

¿Cuál es el error? Corrígelo

En la ficha 3, se coloca una flecha a la izquierda adicional.

El programa corregido se vería de la siguiente forma:



Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

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

1. ¿Puedes reconocer en una experiencia de programación desconectada los conceptos de algoritmo, programa, procesador y programador(a)?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes proponer, ejecutar y verificar una secuencia de instrucciones para realizar una tarea?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a leer los contenidos. Resalta o subraya los términos que no hayas comprendido. Luego, discute con tus compañeras y compañeros de grupo lo que se hizo en cada momento de la actividad y el rol al que correspondía. Si todavía te quedan dudas, consúltale a tu docente.

Los procesadores de los computadores, robots, lavadoras, celulares y muchos otros dispositivos siguen las instrucciones de un programa que ha sido desarrollado por una persona a la que se le conoce como programador o programadora.

Actividad opcional para seguir consolidando aprendizajes

Finaliza haciendo un crucigrama con al menos 6 palabras claves que hayas recordado o aprendido en esta sesión. Crea la cuadrícula y escribe las pistas correspondientes a las palabras horizontales y verticales, pero sin escribir las respuestas. Luego, intercambia crucigramas con alguna compañera o compañero de clase y trata de descubrir las palabras en su juego, a partir de las pistas dadas.

Sesión 2

Aprendizajes esperados

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

- Usar la interfaz del editor de bloques *MakeCode*.
- Utilizar el editor *MakeCode* para crear un programa que muestre números o íconos en pantalla.
- Ejecutar y probar el código creado en el simulador de la *micro:bit*.

Duración sugerida



15% 70% 15%



Material para la sesión

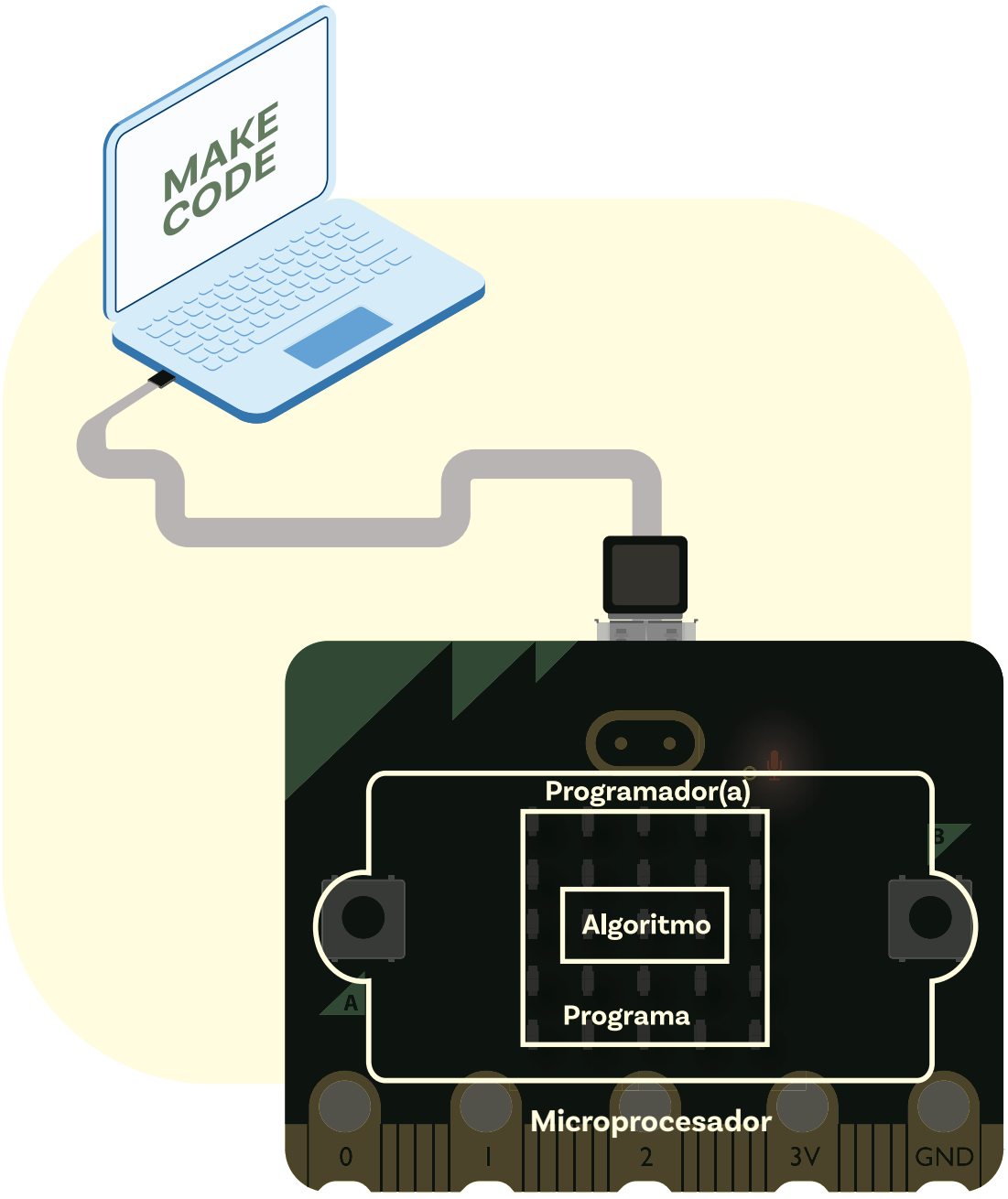
Material por grupo:

- ○ Dispositivo con acceso a *MakeCode*.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

La siguiente imagen te permitirá recordar lo aprendido en la sesión anterior. Obsérvala con atención.



Como seguro notaste en la imagen anterior, al igual que en un rompecabezas, todo lo que has aprendido hasta el momento se conecta. Por ejemplo, un(a) programador(a) necesita crear un algoritmo para así convertirlo en un programa que pueda ser ejecutado por un procesador.

Si en grados anteriores trabajaste con algún lenguaje de programación como *Scratch*, recordarás que lenguajes de este tipo cuentan con bloques que se unen como un rompecabezas y permiten crear animaciones en la pantalla del computador. Si no cuentas con esta experiencia previa o si no recuerdas, no te preocupes, solo observa la *Figura 1*, en la que se presentan los bloques de código en *Scratch* para que el personaje Gato salude diciendo "¡Hola!". De igual manera, puedes ver la animación resultante al ejecutar el programa.

Figura 1. Código en *Scratch* para que Gato diga ¡Hola!



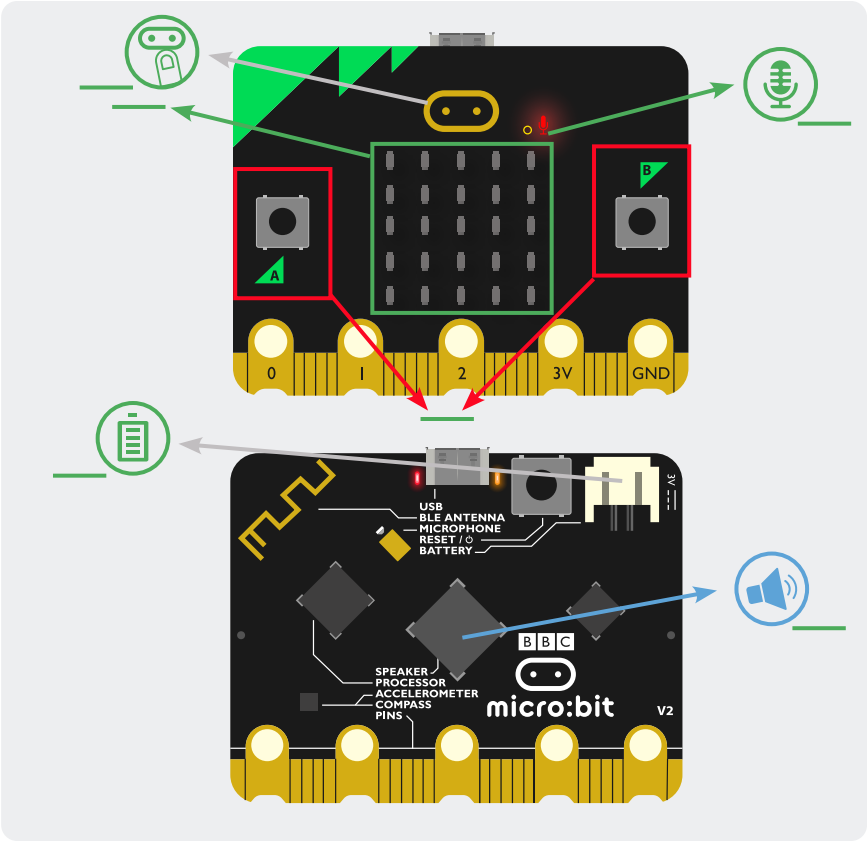


A partir de esta guía, aprenderás un nuevo lenguaje de programación llamado *MakeCode*. Con él, podrás crear nuevos programas para interactuar con la tarjeta *micro:bit*, que es un microcontrolador, como si fuera un computador de bolsillo. La tarjeta cuenta con sensores y una pantalla de LED que te permitirán interactuar con los programas de manera física.

Para construir nuestros programas y permitir que la *micro: bit* los ejecute debemos usar un editor de código nuevo. Este editor incluye un simulador que te permite ver cómo se ejecutará el programa en la tarjeta, ayudándote a visualizar su funcionamiento antes de probarlo en la *micro:bit*.

En la *Figura 2* se muestra una *micro:bit*. Obsérvala cuidadosamente.

Figura 2. Vista frontal y posterior de la *micro:bit*



Los siguientes son algunos de los elementos que integran esta tarjeta. Trata de relacionarlos en la imagen escribiendo la letra correspondiente y, luego, confirma tus respuestas con tu docente.

A. Indicador de batería

B. Sensor de sonido (micrófono)

C. 2 botones programables

D. Altavoz

E. Logo tipo botón táctil

F. Pantalla de LEDs

Enlace

Puedes acceder al editor *MakeCode* en:

También se puede acceder a una versión de *MakeCode* en una aplicación gratuita para el celular.

Nota

MakeCode cuenta con herramientas de accesibilidad como alto contraste y funciona con lectores en pantalla. Consulta a tu docente si necesitas apoyo para hacer uso de estas opciones.

Como notaste, esta tarjeta cuenta con diferentes elementos programables que te permitirán interactuar con el código de manera física. Por ejemplo, podrás mostrar texto e imágenes en la pantalla, apagando o encendiendo LEDs. De igual manera, podrás generar acciones que se inicien al oprimir algún botón o al presionar el logo e incluso ejecutar sonidos desde la tarjeta.

Para construir programas y permitir que la *micro:bit* los ejecute debe usarse el editor de código *MakeCode*. A continuación, vas a explorarlo.

Glosario

LED: es un pequeño bombillo que se puede encender o apagar. La palabra LED viene de su nombre en inglés *(Light Emitting Diode)*, que corresponde a Diodo Emisor de Luz. La *micro:bit* cuenta con 25 LEDs organizados en 5 filas de 5 LEDs cada una. Cada uno de estos LEDs se puede encender o apagar de forma independiente, según se programe.

Micro:bit: dispositivo electrónico que funciona como un computador de bolsillo. Esta tarjeta fue creada por la BBC con el propósito de inspirar la creatividad de niñas, niños, adolescentes y jóvenes y ayudarles a usar sus habilidades de programación para resolver problemas.

Manos a la obra Conectadas

Esta sección corresponde al 85% de avance de la sesión

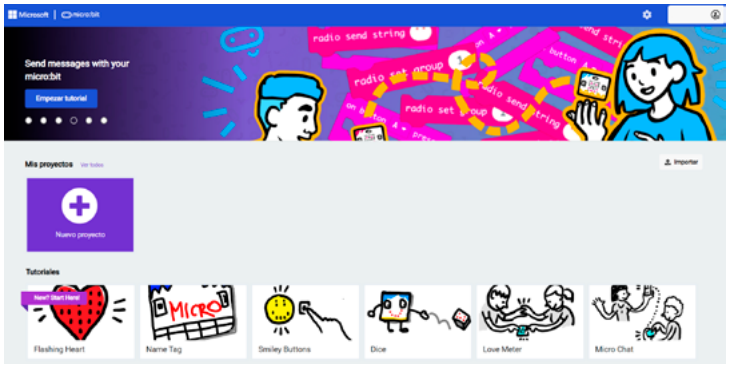
Organízate en parejas siguiendo las instrucciones de tu docente.

Con tu compañera o compañero, ingresen al editor *MakeCode*, utilizando el QR que se presenta en el recuadro enlace. Al hacerlo, verán un pantallazo como el que aparece en la *Figura 3*.

Figura 4. Nuevo proyecto



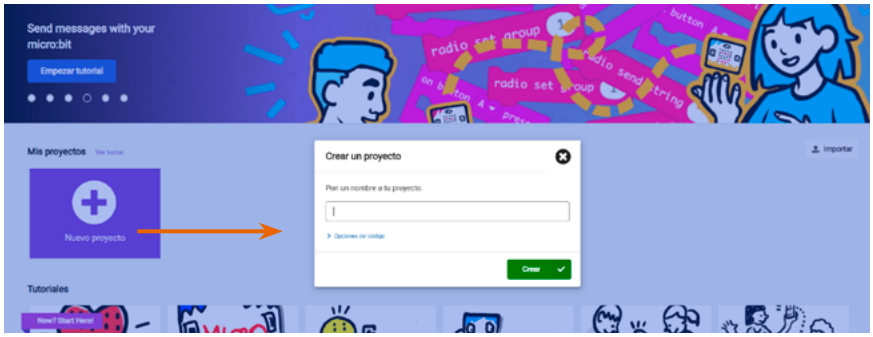
Figura 3. Pantalla inicial *MakeCode*



Luego, den clic al botón que dice “Nuevo proyecto”, ver *Figura 4*.

Les saldrá una ventana emergente que dice “Crear un proyecto”, como se ve en la *Figura 5*. Se tiene un campo para que digiten el nombre que le darás al programa. Escriban “Mi primer programa”, y luego den clic en el botón “Crear”.

Figura 5. Nombrar el proyecto



Deberán estar viendo un pantallazo como el de la *Figura 6*.

Figura 6. Pantallazo nuevo proyecto

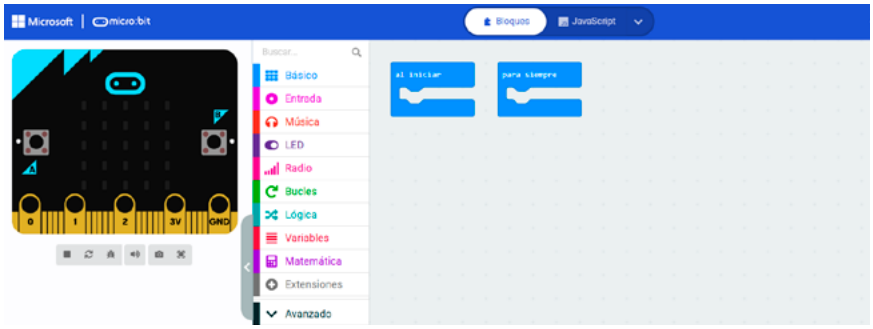
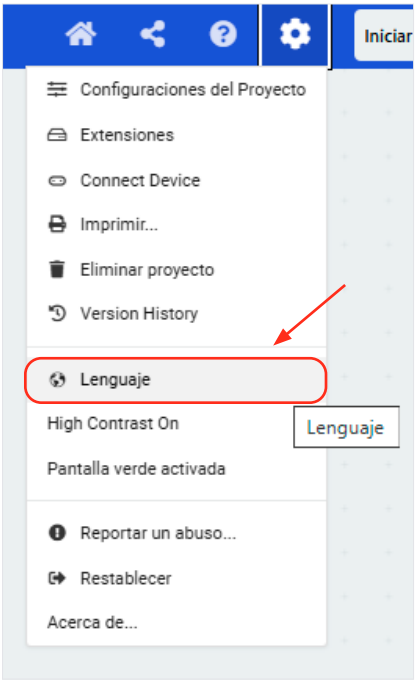


Figura 7. Ajustes



Si el editor no te aparece en español, puedes dar clic en la ruedita o engranaje de ajustes que aparece en la esquina superior derecha, y elegir la opción "Lenguaje", como se ve en la *Figura 7*.

Elige la opción "Español (España)" que aparece disponible allí, como se puede ver en la *Figura 8*.

Figura 8. Selección lenguaje

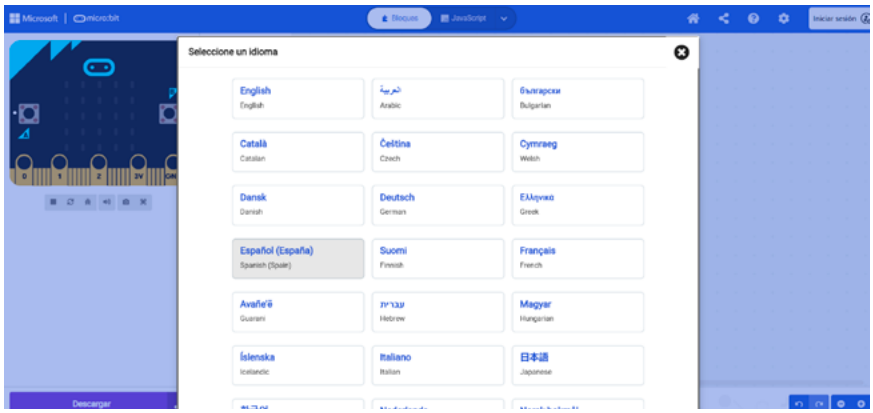
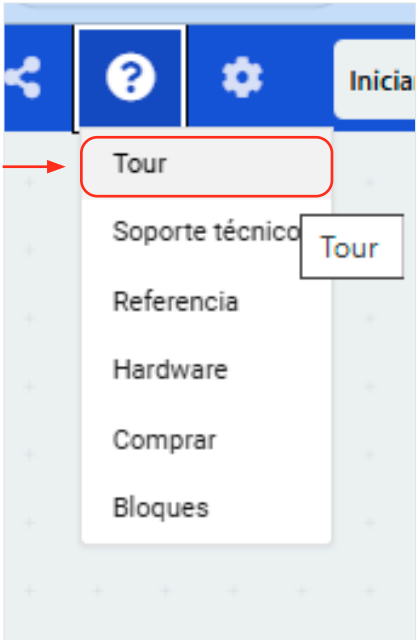


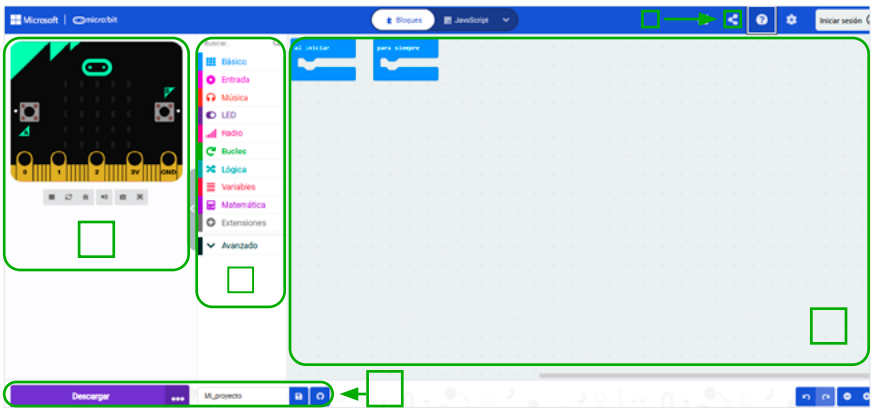
Figura 9. Tour de reconocimiento de la plataforma



Realicen un tour de reconocimiento del editor para identificar 5 elementos claves de la interfaz: 1) el simulador, 2) la caja de herramientas, 3) el área de programación, 4) el enlace para compartir, y 5) el botón de descarga. Pueden acceder al tour, dando clic en el botón de ayuda y eligiendo la opción "Tour", como se presenta en la *Figura 9*.

Ahora, en la *Figura 10* escriban el número que corresponde a cada área del editor. Antes de continuar, confirmen las respuestas con su docente.

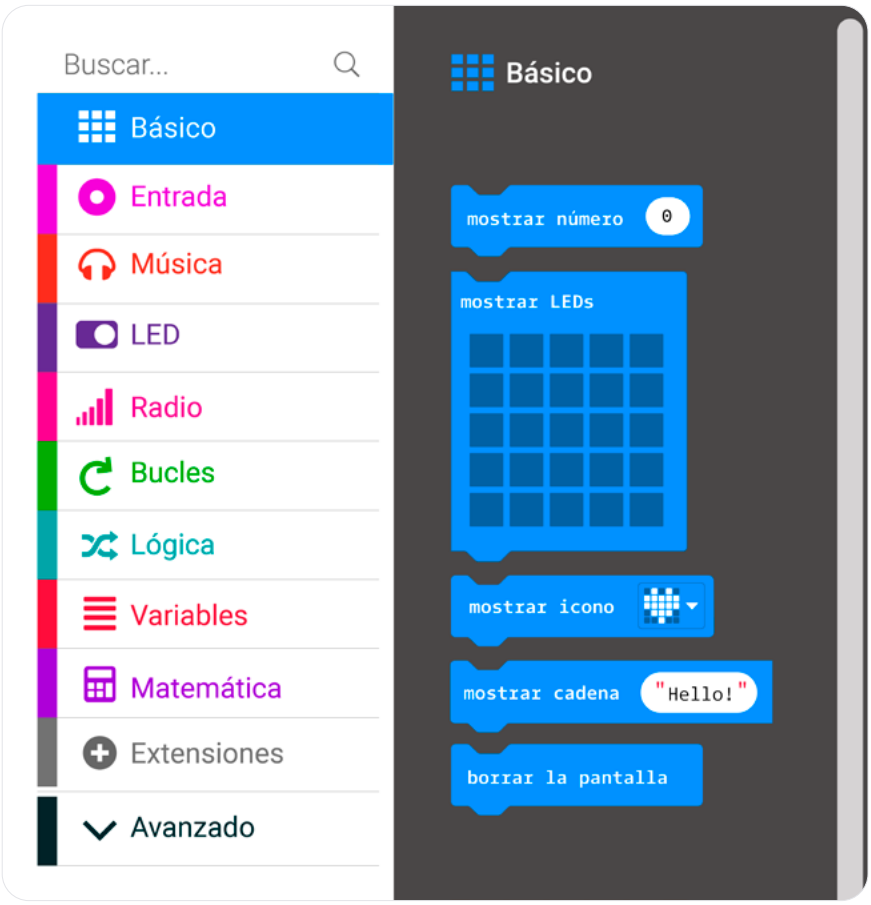
Figura 10. Partes del editor *MakeCode*



¡Ya pueden empezar a crear su primer programa! Sigan estas instrucciones paso a paso.

1. Den clic en la caja de herramientas y elijan la opción “Básico”, ver *Figura 11*. Al hacerlo les aparecerán varios bloques, cada uno de los cuales representa una instrucción, al igual que las flechas en la actividad que se realizó en la sesión anterior.

Figura 11. Bloques básicos de *MakeCode*



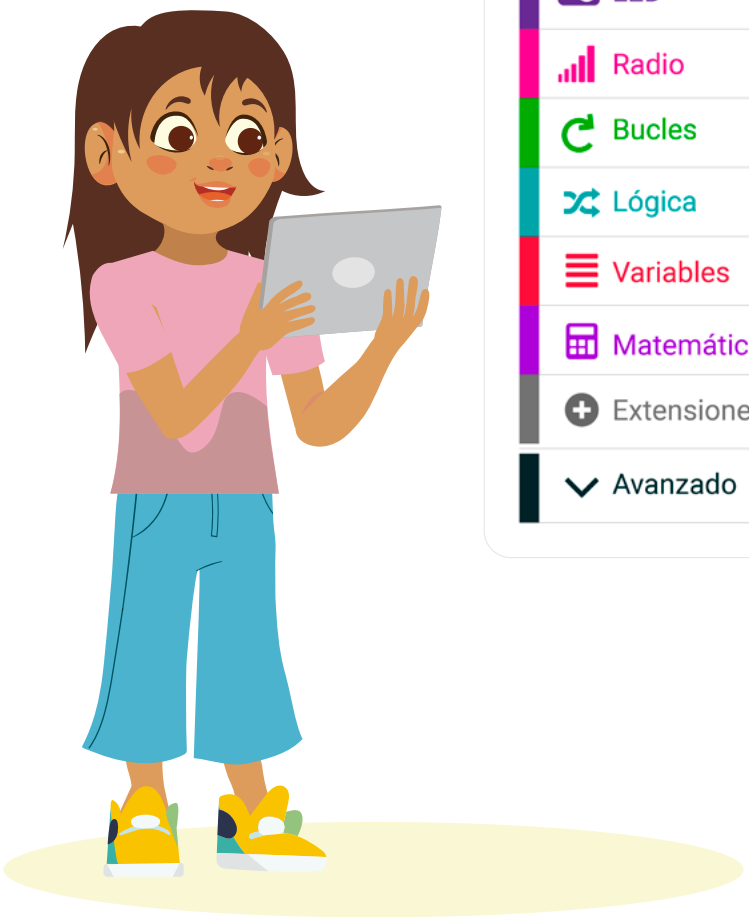


Figura 12. Código para mostrar un número

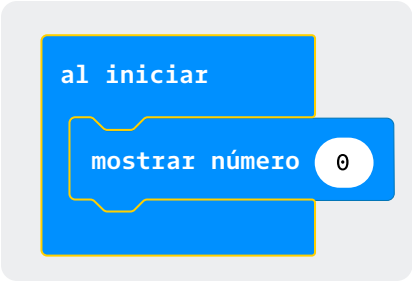


Figura 13. Código para mostrar un ícono



Figura 14. Código para crear un ícono encendiendo cada LED de forma independiente



2. Ubiquen el bloque **mostrar número 0** dentro del bloque **al iniciar**, y observen lo que sucede en la pantalla LED del simulador de la *micro:bit*. ¿Observaron el 0?, ver *Figura 12*.

3. La *micro:bit* puede mostrar números y también íconos. Para ensayarlo, repliquen el programa que aparece en la *Figura 13*. Esto pueden hacerlo tomando los bloques que requieren de la sección "Básico" que aparece en la caja de herramientas.

4. Revisen que los bloques hayan encajado perfectamente. ¿Qué creen que pasaría si algún bloque no encaja? Exploren lo que sucede y prepárense para comentar su hallazgo con el resto de la clase, según lo indique su docente.

5. Como vieron en la imagen inicial de la *micro:bit*, esta tarjeta contiene unos dispositivos que emiten luz, llamados LEDs. Ahora verán que entre el grupo de bloques "Básico" también hay uno que permite decidir qué LED encender. Para esto pueden usar el bloque **mostrar LEDs**. Agreguen este bloque para hacer la X más pequeña, ver *Figura 14*.

6. ¿Lo lograron? Si es así, ¡felicitaciones, ya crearon su primer programa! Si todavía no lo han logrado, es el momento de hacer una pausa para revisar.

   A. Revisen lo que hicieron desde el inicio del proceso.

   B. Comparen lo que hicieron con lo que haya hecho otro grupo.

   C. Si aún no lo logran, pídanle ayuda a su docente.

7. Ahora van a mostrar una secuencia de imágenes utilizando lo que han aprendido hasta este momento. Para esto deberán añadir más bloques **mostrar LEDs** y cambiar el orden de algunos otros bloques. Además, tendrán que reemplazar el bloque **al iniciar**, por el bloque **para siempre,** ver *Figura 15*. De esta forma, las imágenes se repetirán una y otra vez, sin detenerse.

Figura 15. Código para mostrar una secuencia de imágenes



Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

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

1. ¿Puedes reconocer algunos elementos de la interfaz del editor de bloques *MakeCode*?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes utilizar el editor *MakeCode* para crear un programa que muestre números o íconos en pantalla?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

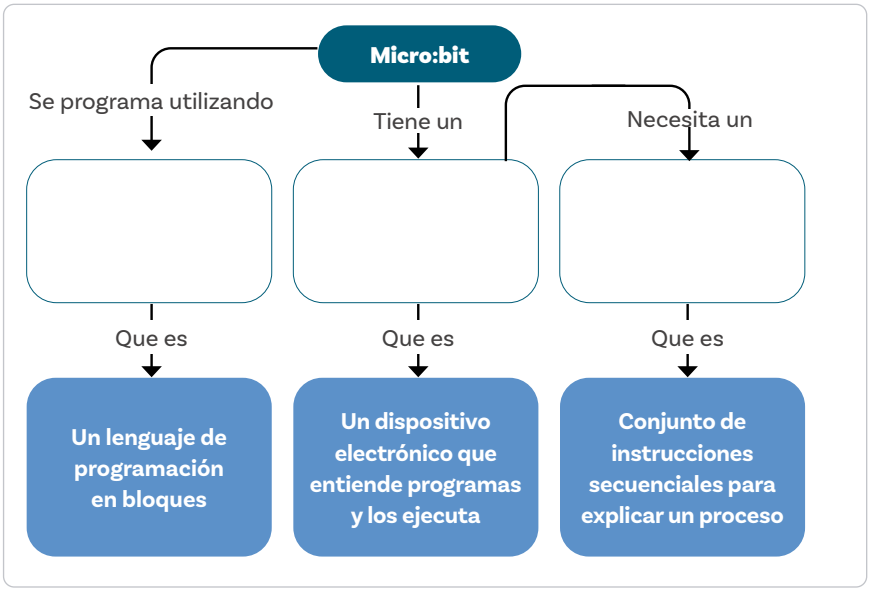
3. ¿Puedes visualizar el código creado en el simulador de la *micro:bit*?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si respondiste "Parcialmente" o "Aún no", regresa al editor *MakeCode* y verifica las respuestas de los diferentes elementos que te presenta este lenguaje al realizar el tour. Luego, sigue los pasos sugeridos para crear el código que aparece en la *Figura 7*. Cuando lo hayas hecho, observarás que los íconos que agregaste generan un efecto visual, pues pareciera que la X crece y luego desaparece, de forma repetitiva. En la siguiente sesión de clase aprenderás más sobre secuencias de imágenes como esta. Si todavía te quedan dudas, coméntalas con tu docente.

Finaliza completando los espacios en blanco en el siguiente gráfico de anclaje. Agrégales imágenes que consideres adecuadas para ayudarte a recordar mejor.

Gráfico de anclaje. Sesión 2, guía 1, grado 5°



Actividad opcional para seguir consolidando aprendizajes

Ahora, imagina que le vas a contar a una persona de tu familia o alguna amiga o amigo tuyos lo que hiciste y aprendiste hoy. Completa las siguientes frases:

En clase el día de hoy _____

_____

_____

Aprendí que _____

_____

_____

Lo que más me gustó fue _____

_____

_____

Lo más difícil fue _____

_____

_____

Sesión 3

Aprendizajes esperados

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

- Crear un programa que genere una animación sencilla en *MakeCode* basada en secuencias de íconos.
- Aplicar la descomposición a la creación de animaciones con ayuda de la *micro:bit*.

Duración sugerida



33% 34% 33%



Material para la sesión

Material por grupo:

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

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 33% de avance de la sesión

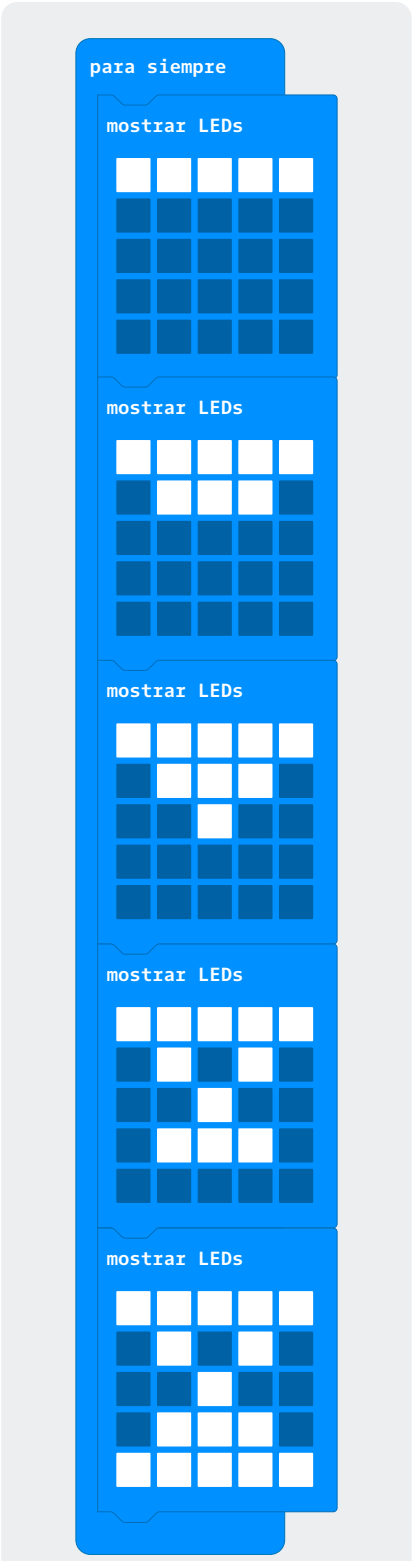
Las animaciones son como películas o dibujos que se mueven en la pantalla. ¿Recuerdas esos dibujos que pasan muy rápido en la tele y hacen que los personajes se muevan? ¡Eso es una animación! Pero, ¿cómo se hacen? Bueno, imagina que tienes un montón de dibujos, cada uno un poquito diferente del anterior. Si los pasas muy rápido uno detrás del otro, ¡parecerá que los dibujos cobran vida y se mueven! Eso es lo que hace una animación: muestra varias imágenes, una después de la otra, muy rápido.

Figura 1. Ejemplo de animación



Ahora, ¿cómo las hacemos en un computador? Ahí es donde entra la programación. Imagina que cada dibujo tiene un número. Con la programación, se le dice al computador en qué orden se quiere que muestre esos dibujos y también se le indica por cuánto tiempo debe mostrar cada imagen. Es como si se le dijera: "muestra el dibujo número uno por un segundo, luego el dibujo número dos por otro segundo, y así sucesivamente". Así, el computador une todas esas imágenes en orden y las muestra tan rápido que parece que los dibujos están cobrando vida. ¡Es como magia!

Figura 2. Código para crear una animación



Manos a la obra

Conectadas

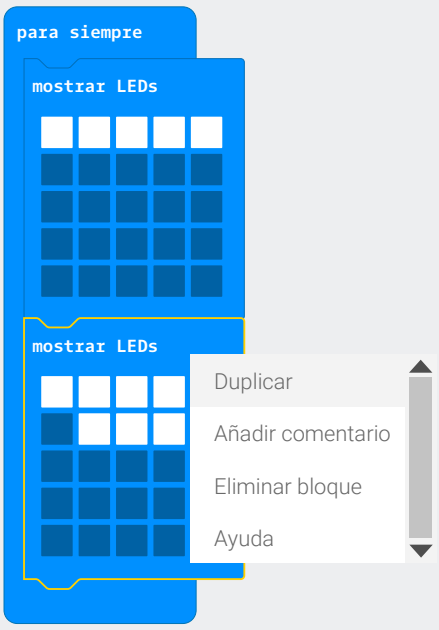
Esta sección corresponde al 67% de avance de la sesión

En la sesión anterior ya tuviste la primera oportunidad de interactuar con el entorno de programación en bloques *MakeCode*. Ya pudiste mostrar un número y también mostrar unos íconos con la pantalla de LEDs. ¿Qué tal si ahora generas una pequeña animación?

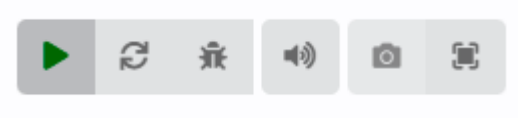
Organízate con una compañera o compañero, según las indicaciones de tu docente, e ingresen a *MakeCode*. Pueden guiarse por las instrucciones de la sesión anterior para crear un nuevo proyecto y darle nombre. En este caso, nombren el programa “animacion1”.

Luego, dentro del bloque **Para siempre**, ingresen bloques **Mostrar LEDs** y márquenlos como se indica en la *Figura 2*. Pueden ubicarse sobre un bloque, dar clic derecho y elegir la opción “Duplicar bloque”, que se activa en el menú contextual que aparece, tal y como se presenta en la *Figura 3*.

Figura 3. Menú contextual que aparece al dar clic derecho sobre un bloque

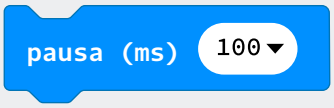


Una vez tengan listo el programa, ejecútenlo dando clic en el triángulo que aparece debajo del simulador.



¿Qué observan? ¿Pueden notar que pareciera que se muestra en la pantalla un reloj de arena animado?

1. Ahora, en la caja de herramientas, conjunto de bloques básico, busquen el siguiente bloque.



Y añádanlo al programa.

2. ¿Descubrieron para qué sirve esta nueva instrucción? Compartan sus respuestas.

3. Con ayuda de los bloques vistos, modifiquen la animación del reloj de arena para que sea más lenta o rápida. Tengan en cuenta que en el bloque pausa la medida de 100 (ms) es un espacio de tiempo muy pequeño. Si se quiere dar una pausa de un segundo, se debe indicar que la pausa será de 1000 y si se eligen 500, la pausa será de la mitad de un segundo.

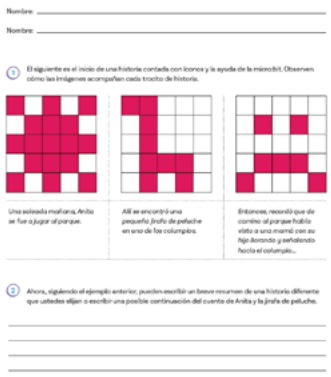
4. Agreguen una pausa después de cada bloque mostrar LEDs para completar la animación.

Ya que han finalizado este ejercicio, es hora de que intenten hacer su propia animación, eligiendo o creando las imágenes que prefieran para contar una historia corta (5 a 9 íconos). El *Anexo 3.1* les ayudará a organizar sus ideas. Sigan las instrucciones de su docente para desarrollarlo.

Cuando hayan terminado de trabajar con el anexo, piensen en las diferentes etapas que implica crear una animación como la que hicieron y hagan un listado de esas tareas. De esta forma están descomponiendo una tarea grande en partes pequeñas.

Anexo

Anexo 3.1



3 páginas

Creación de animaciones en *MakeCode*

a. Pensar en las posibles historias que se podrían animar.

b. _____

c. _____

d. _____

e. _____

Glosario

Ícono: símbolo o imagen sencilla que representa algún objeto.

Milisegundos (ms): unidad de tiempo que corresponde a una milésima fracción de segundo, es decir, 1 segundo dividido en 1000. Esta es la unidad de tiempo que utiliza *MakeCode* para generar pausas dentro de un programa.

Descomposición: habilidad para dividir una tarea grande en otras más pequeñas, a fin de darle solución.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, revisa los aprendizajes de la sesión, respondiendo las siguientes preguntas.

1. ¿Puedes crear un programa que genere una animación sencilla en *MakeCode*, basada en secuencias de íconos?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

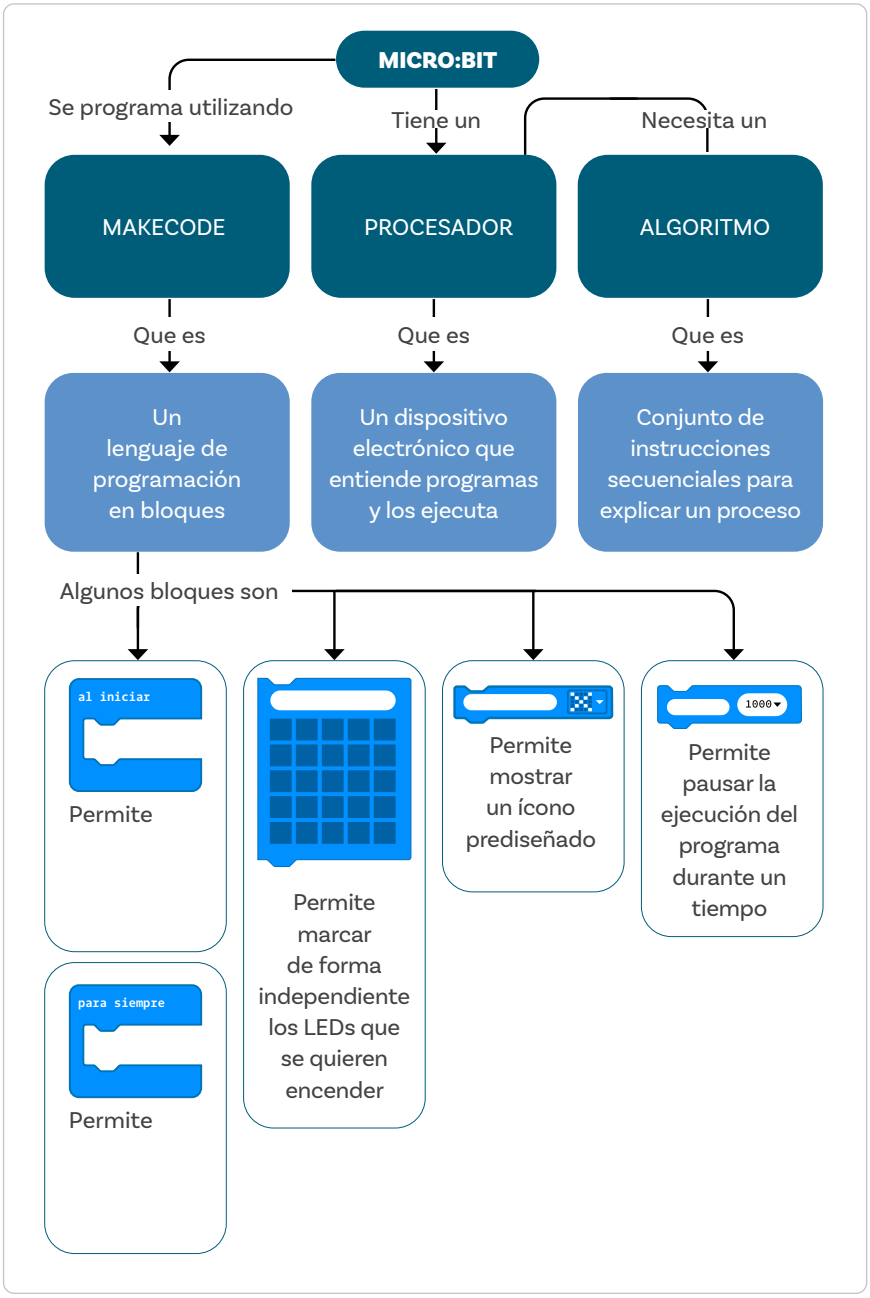
2. ¿Puedes aplicar la descomposición a la creación de animaciones con ayuda de la *micro:bit*?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

En caso de que hayas respondido "Parcialmente" o "Aún no" a alguna de las preguntas anteriores, devuélvete y revisa el glosario de esta sesión. Después, revisa el trabajo que tu compañera o compañero y tú hicieron al desarrollar el *Anexo 3.1.* También, mira las tareas que listaron como pasos requeridos para crear una animación en *MakeCode*. ¿Hay algo de lo que hicieron con lo que sientas que todavía tienes dudas? Chárlalo con la persona con la que hiciste la actividad. Si después de esta conversación todavía sientes que necesitas apoyo, pide ayuda a tu docente.

Completen los espacios en blanco en el siguiente gráfico anclaje:

Gráfico de anclaje. Sesión 3, guía 1, grado 5°



Prepárense para compartir sus respuestas con toda la clase, siguiendo las indicaciones de su docente.

Actividad opcional para seguir consolidando aprendizajes

Te proponemos la siguiente actividad para seguir practicando la creación de animaciones con la *micro:bit*. Crea un pequeño programa para representar una historia diferente o incluso un proceso por medio de íconos. Piensa que este programa pueda servirle como ayuda de estudio a otras personas. Por ejemplo, ¿qué tal si haces una animación para enseñar y aprender las etapas del ciclo del agua?

Después de realizar esta segunda animación, sigue las instrucciones de tu docente para organizarte en grupos de 3 a 5 personas.

1. Comparen sus listados de tareas para la creación de animaciones en *MakeCode* y luego discutan estas preguntas: ¿Tienen el mismo número de tareas listadas? ¿Alguna persona escribió algo que las demás no habían considerado u olvidó incluir una de las tareas? Después de haber comparado listados, ¿hay algo que cambiarían en su propia lista?

2. Como grupo, escriban una conclusión sobre lo que se debe tener en cuenta a la hora de descomponer una tarea en otras más pequeñas. Asegúrense de que pueden escribirla de modo que quepa en una nota adhesiva.

Sesión 4

Aprendizajes esperados

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

- Predecir los resultados de programas sencillos.
- Reconocer cuando el resultado de un programa no es el esperado.
- Depurar programas según un objetivo planteado.

Duración sugerida

33% 34% 33%



Material para la sesión

Material por grupo:

- Dispositivo con acceso a *MakeCode*.

Figura 1. Ejemplo de animación con errores



Lo que sabemos, lo que debemos saber

Esta sección corresponde al 33% de avance de la sesión

En la clase anterior pudiste ver una animación en el simulador de *MakeCode*. Entonces, ya sabes que esta herramienta del editor te permite ejecutar las instrucciones que has programado con el lenguaje de bloques y, de este modo, **validar** que tu solución realiza lo que deseas que haga.

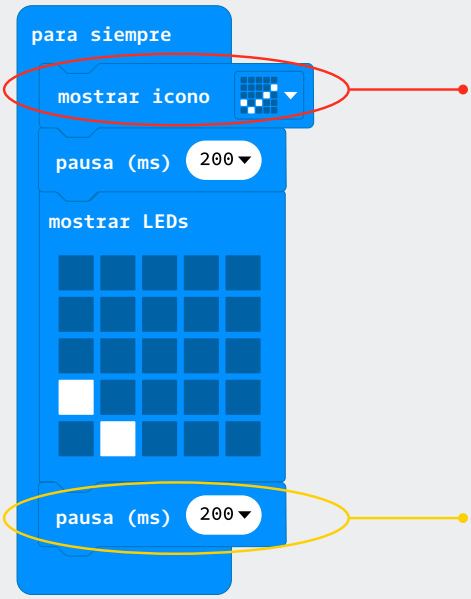
Si al completar el proceso anterior encuentras que hay errores, entras en el proceso de **depuración**, es decir, los corriges para que tu programa funcione correctamente.

Ahora vas a aplicar lo aprendido haciendo un programa en *MakeCode*, validando que realice lo que deseas y, luego, corrigiéndolo si es necesario.

Para lo anterior, observa el programa de la *Figura 1*, luego ingresa al editor y prográmalo. Después, valida que realice lo siguiente:

- ◯ Mostrar de forma permanente (para siempre) parte de un signo de verificación o chulo (✓).
- ◯ Esperar 200 ms y mostrar el chulo completo.
- ◯ Luego, esperar 250 ms.

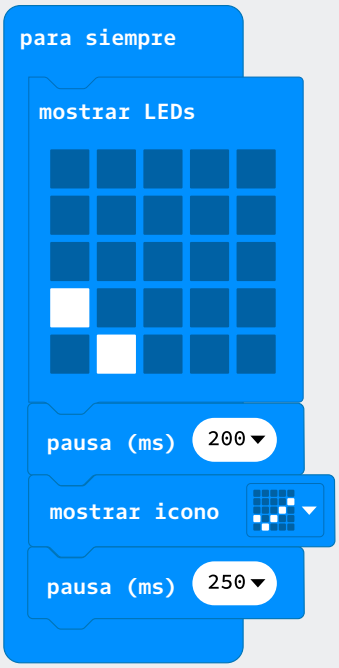
Al validar, seguro encontraste que realiza lo deseado, pero con algunos errores. Observa a continuación:



Comienza con el chulo completo, pero debería comenzar con la porción de este.

La pausa que hay configurada es de 200 ms y se necesita de 250 ms.

Figura 2. Código de animación depurado



Es necesario, entonces, **depurar** el programa, reubicando los bloques mostrar ícono y mostrar LEDs, y cambiando el valor de la segunda pausa. Cuando hayas hecho estos cambios, tu programa se verá como el de la *Figura 2*.

Glosario

Validar: verificar que el programa hace lo que debe hacer

Depurar: identificar y corregir errores o fallos en un programa para que funcione correctamente.

Figura 3. Programa 1 para depurar

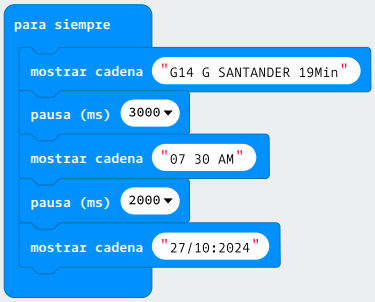
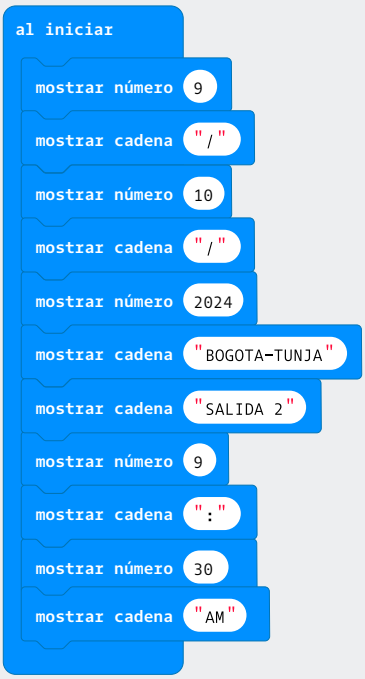


Figura 4. Programa 2 para depurar



Manos a la obra Conectadas

Esta sección corresponde al 67% de avance de la sesión

Ya has visto cómo **validar** y **depurar** un programa. ¡Es tiempo de que apliques lo aprendido realizando algunas tareas!

Sigue las instrucciones de tu docente para trabajar con alguna compañera o compañero. De forma individual deberán elegir un programa diferente para analizar (*Figura* 3 o 4). Sin programarlo en el editor todavía, deberán imaginar y escribir lo que piensan que hará ese programa al ejecutarlo en el editor.

Programa 1 (Figura 3)

Escribe tu explicación aquí: _____

_____

_____

_____

_____

_____

Programa 2 (Figura 4)

Escribe tu explicación aquí: _____

_____

_____

_____

_____

_____

_____

Compartan lo que escribieron sobre cada uno de los programas que analizaron de forma individual.

Ahora sí, ingresen a *MakeCode* para verificar lo que dijeron que sucedería. Primero hagan un programa, ejecútenlo y coloquen en la tabla una X si hace lo que dijeron que pasaría o no. Luego, hagan lo mismo con el programa dos.

¿El programa hizo lo que creyeron que pasaría?		
Programa	**Sí**	**No**
Programa 1		
Programa 2		

Estos dos programas que analizaron tenían algo en común: ambos comunicaban información asociada a medios de transporte. El primero mostraba información de un Sistema Integrado de Transporte[2], como la de una ruta de TransMilenio en la ciudad de Bogotá, y el segundo mostraba información de un sistema de transporte intermunicipal[3]. Como los mensajes son emitidos por empresas diferentes, cada una incluye información de forma distinta en sus comunicados.

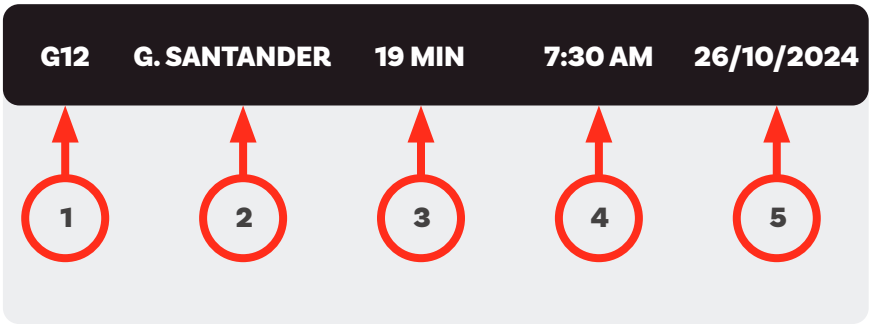
Su misión ahora es validar que los programas respeten las siguientes reglas, y, si es necesario, depurarlos en el entorno de *MakeCode*.

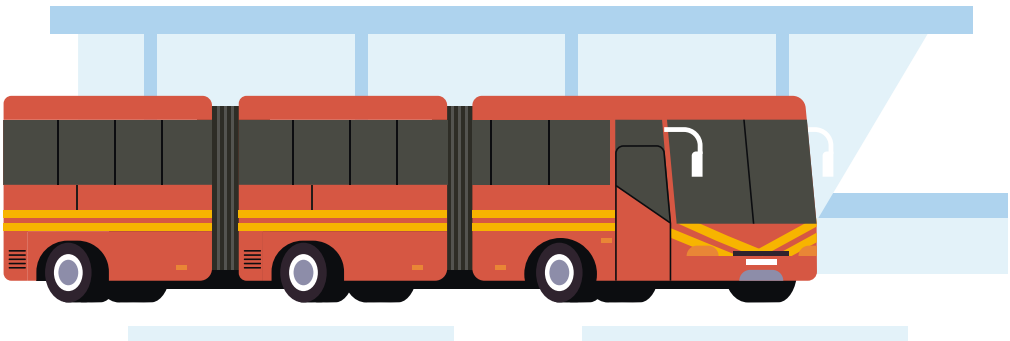
2. Un Sistema Integrado de Transporte es un conjunto de medios de transporte que trabajan como un único equipo para ayudar a las personas a movilizarse de un lado a otro en una ciudad, sin dificultades. Estos sistemas tienen diferentes medios de transporte, por ejemplo, metro, tranvía, cable aéreo, bus e incluso bicicletas de uso público.
3. Es decir, un servicio terrestre de transporte de pasajeros de una ciudad o municipio a otro. Los buses que viajan de una ciudad a otra forman parte de este sistema.

El Programa 1 debe:

1. Mostrar una letra y un número que indiquen el nombre de la ruta.
2. Mostrar el nombre de la estación que corresponda a la parada final del bus.
3. Mostrar el tiempo que tarda en llegar la ruta a su destino en la estación.
4. Mostrar la hora actual.
5. Mostrar la fecha actual.

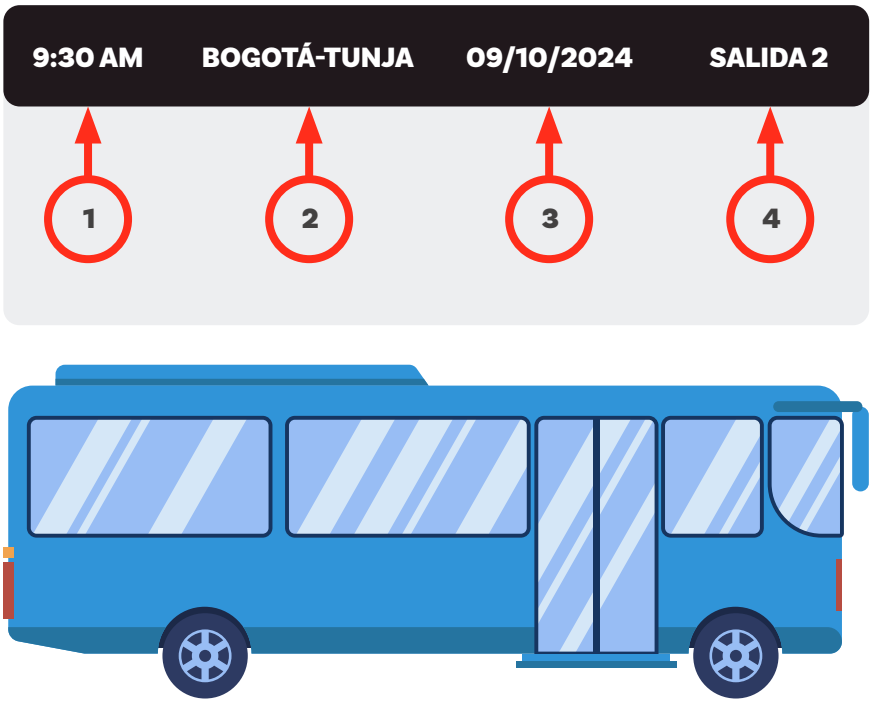
La imagen a continuación te muestra cómo debería verse esta información.





El Programa 2 debe:

1. Mostrar la hora actual.
2. Mostrar la ciudad de origen y la de destino.
3. Mostrar la fecha actual.
4. Mostrar la puerta de salida donde se espera el bus.



Sigan las instrucciones de su docente para comparar sus programas depurados con los realizados por otros grupos. ¿Hay diferencias entre los cambios que hicieron?

Ahora, como grupo, discutan:

Ya que han aprendido a simular la programación de mensajes para letreros LED, ¿han visto mensajes o anuncios publicitarios que utilicen letreros luminosos como estos? Si es así, ¿dónde los han visto?

¿Qué podría pasar si hay un error en la programación de los letreros de las rutas de transporte?

Considerando lo realizado en la clase, ¿por qué es importante validar y depurar los programas?

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

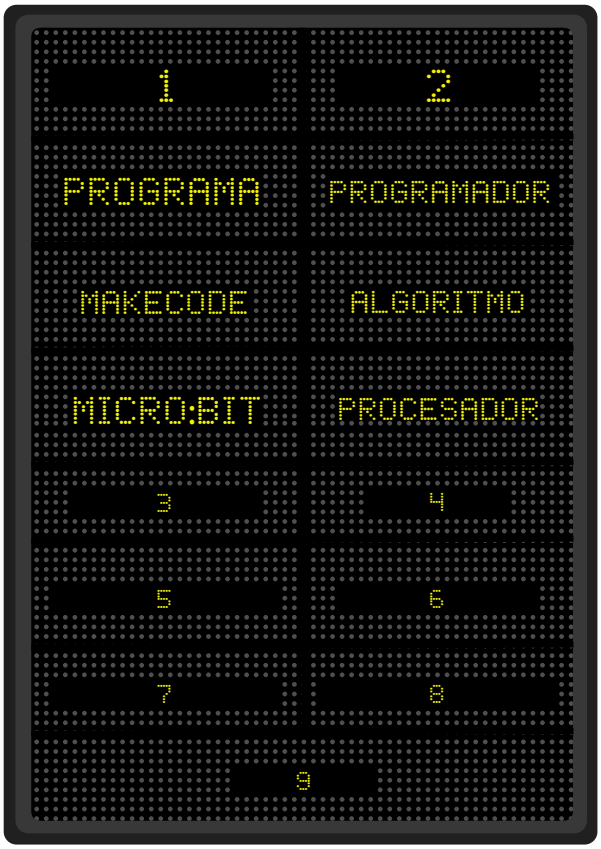
Pistas:

1. Corregir los errores que hace que un programa no funcione.
2. Verificar que un programa hace lo que debe hacer.
3. Nombre del bloque que permite ejecutar las acciones de un programa una y otra vez hasta que se detenga la simulación.
4. Nombre del bloque que permite ejecutar el código introducido solo una vez, cuando se inicia la simulación.
5. Nombre del bloque que permite mostrar números en el simulador.
6. Nombre del bloque que permite encender de forma independiente los LEDs.
7. Nombre del bloque que permite mostrar un ícono prediseñado del editor.
8. Nombre del bloque que permite mostrar texto.
9. Nombre del bloque que permite detener la ejecución del programa por el tiempo que se indique.

Trabajarás de forma individual otra vez.

La *Figura 5* es un gráfico de anclaje que te ayudará a recordar aspectos relevantes aprendidos hasta ahora. Para completarlo, lee atentamente las pistas de la página anterior, descubre las palabras secretas correspondientes y escríbelas en los espacios vacíos identificados por los números.

Gráfico de anclaje. Sesión 4, guía 1, grado 5°



1. _____
2. _____
3. _____ _____
4. __ _____
5. _____ _____
6. _____ ____
7. _____ _____
8. _____ _____
9. _____ ____

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

1. ¿Puedes predecir los resultados de programas sencillos?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2. ¿Puedes reconocer cuando el resultado de un programa no es el esperado?

- ◯ Sí
- ◯ Parcialmente
- ◯ No

3. ¿Puedes depurar programas según un objetivo planteado?

- ◯ Sí
- ◯ Parcialmente
- ◯ No



Si tus respuestas fueron "Parcialmente" o "Aún no" mira tus notas sobre la actividad de análisis de los 2 programas. Luego, revisa lo que hiciste junto a tu compañera o compañero en *MakeCode*. Al detectar y corregir errores en las instrucciones o códigos, validaron y depuraron los programas. Comparte las dudas que aún tengas con tu docente, quien te ayudará a aclararlas.

Sesión 5

Aprendizajes esperados

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

- Utilizar entradas y salidas booleanas en un programa en *MakeCode*.
- Codificar mensajes representados de forma visual no alfabética.
- Crear, validar y depurar un programa para que transmita un mensaje visual según especificaciones dadas.

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

Hasta el momento has trabajado con los LEDs de la *micro:bit*, representando íconos que tienen algún significado. Por ejemplo, una X o un reloj de arena, que pueden significar error y el transcurso del tiempo, respectivamente. A estas representaciones, entre otras que se pueden crear, en programación se les conoce como **codificar**.

Ingresa al editor *MakeCode* y crea un nuevo proyecto.

En esta sesión aprenderás a codificar información haciendo uso de un nuevo bloque llamado al presionarse el botón. Lo encontrarás en la caja de herramientas, en la sección "Entrada", como se muestra en la *Figura 1*.

Escribe en el editor el programa que encuentras en la *Figura 2*.

En el simulador, identifica los dos botones que tiene la *micro:bit*, los botones A y B. ¿Qué sucede si oprimes el botón A?

Figura 1. Caja de herramientas, sección "Entrada"



Figura 2. Programa para mostrar un ícono al presionar un botón



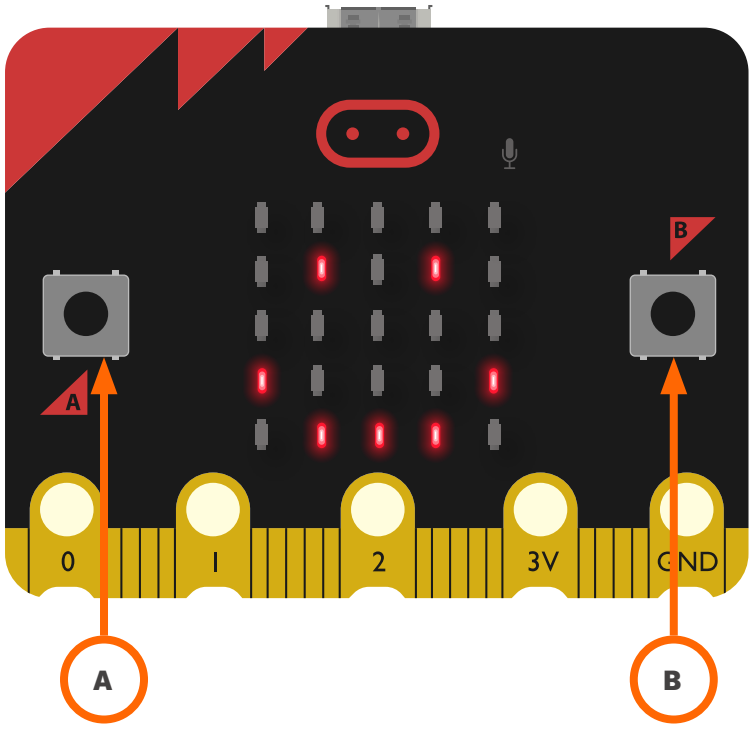
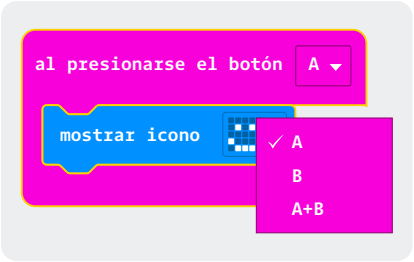


Figura 3. Elegir botón para activar instrucciones

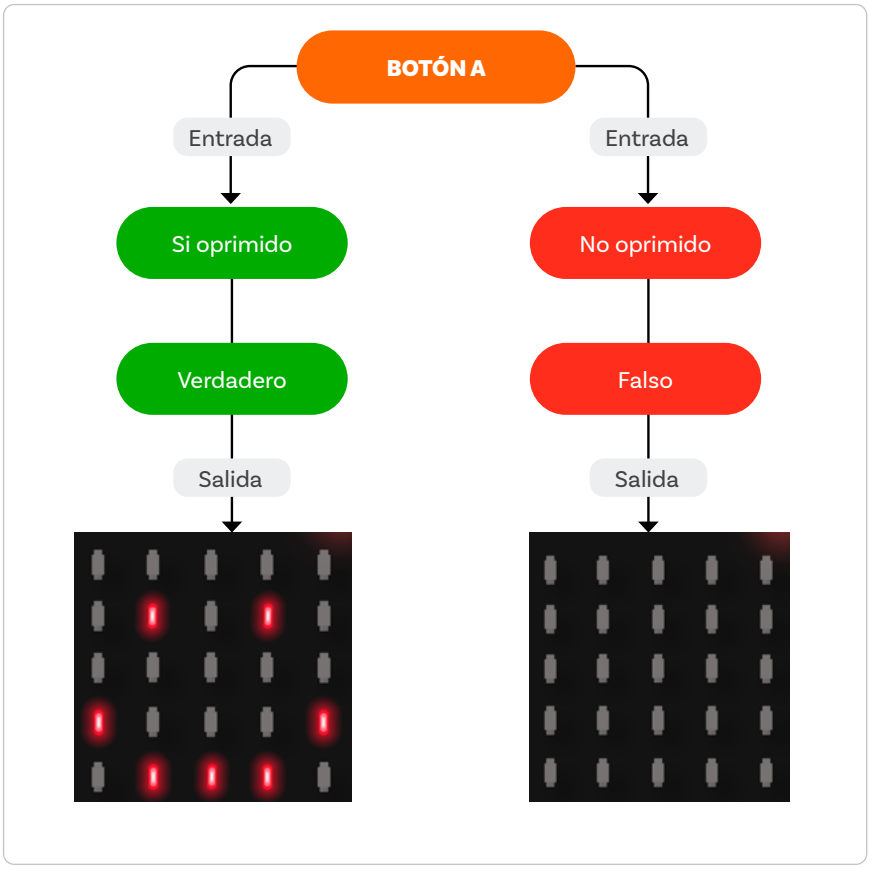


Agrega otro bloque Al presionar el botón, y explora la lista de opciones. Tu misión será lograr que la *micro:bit* muestre la imagen de una cara triste cuando se presione el botón B, como indica la *Figura 3*. ¿Lo conseguiste?

En la *micro:bit* a los botones (A y B) se les conoce como **entradas** de información ya que, si se oprime el botón, le estás indicando a la *micro:bit* que debe ejecutar una acción. En el ejercicio de la cara feliz, al momento de oprimir el botón A los LEDs se encienden, mostrándonos el ícono correspondiente. Entonces, si tenemos una entrada de información al oprimir el botón, también tendremos una salida de información cuando se encienden los LEDs. De esta manera, a los LEDs en la *micro:bit* se les conoce como **salidas** de información.

Los botones y los LEDs se consideran entradas y salidas **booleanas**. Observa el gráfico de anclaje.

Gráfico de anclaje. Sesión 5, guía 1, grado 5°



En programación, un booleano es una variable que solo puede tener dos valores: verdadero o falso. Este tipo de variable se usa para representar situaciones en las que algo puede ocurrir o no ocurrir.

Por ejemplo, con el botón A de la *micro:bit* pueden suceder dos cosas: oprimirlo o no oprimirlo. Si la indicación es que al oprimir el botón A se muestre la cara feliz, entonces cuando el botón está presionado el valor es verdadero. En cambio, cuando no se oprime, no aparece la cara feliz y el valor es falso.

Glosario

- **Codificar:** proceso de transformar un tipo de información en otra.
- **Booleana:** información que solo puede tener dos valores: verdadero o falso.
- **Entrada:** información que ingresa a un sistema y que puede influir en su funcionamiento o en el resultado final.
- **Salida:** resultado o efecto producido por un sistema después de procesar la entrada.



Enlace

Si lo deseas, puedes ver el video de este QR para que aprendas un poquito más sobre el sistema Braille.

Manos a la obra

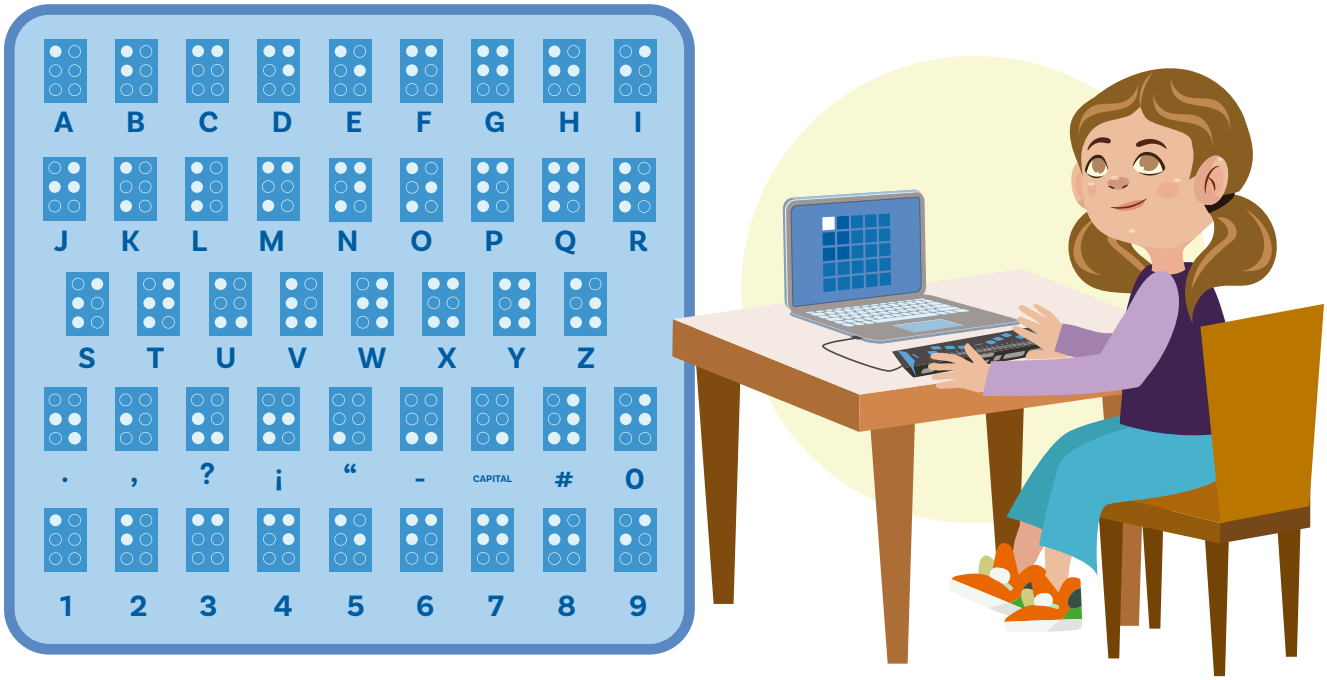
Conectadas

Esta sección corresponde al 85% de avance de la sesión

¿Te has preguntado alguna vez cómo leen y escriben las personas con discapacidad visual? Si no lo sabes, en esta actividad conocerás su medio de comunicación escrita, el Braille. El lenguaje Braille es un sistema de lectura que se siente al tacto. Cada letra, número o signo de puntación se representa por medio de puntos pequeños al relieve que se pueden "leer" al tocarlos con las yemas de los dedos.

Hoy en día existen tecnologías que ayudan a las personas que no pueden ver a leer textos digitales. Una de estas tecnologías es la Línea Braille (*Figura 4*) que tiene pequeñas cajitas de 6 u 8 puntos que suben y bajan para formar letras. La línea Braille se conecta a la pantalla de un celular o computador y a medida que cambian las letras en la pantalla, las letras en Braille también cambian para que la persona pueda leerlas con sus manos.

Figura 4. Alfabeto y línea



Tu misión será programar un simulador de línea Braille que muestre, por medio de los LEDs de la *micro:bit*, un mensaje positivo y motivador codificado en este sistema de lectura.

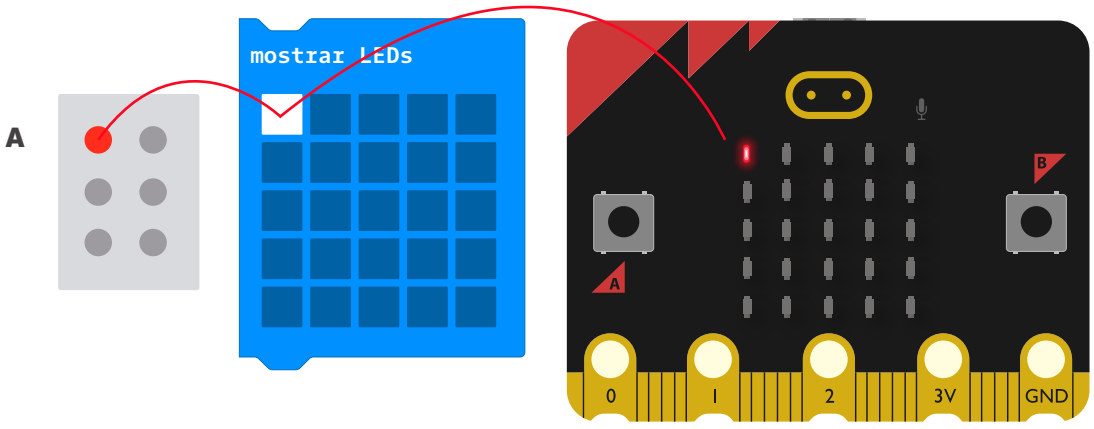
Organízate con una compañera o compañero siguiendo las indicaciones de tu docente.

1. Elijan el mensaje que desean transmitir. Asegúrense de que sea corto (2 a 3 palabras preferiblemente) y que pueda hacer feliz a quien lo descifre. Por ejemplo, piensen frases como estas: “Eres maravillosa”, “Ánimo, puedes lograrlo” o “Sonríe cada día”.

2. En una hoja de papel, codifiquen este mensaje, reemplazando cada letra por los puntos que la representan en el alfabeto Braille.

3. Ingresen a *MakeCode* y creen un nuevo proyecto. Al crear el programa, utilicen el LED superior izquierdo de la *micro:bit* como el primer círculo de la parte superior izquierda de cada letra o símbolo del abecedario en Braille.

Figura 5. Uso del bloque mostrar LEDs para representar una letra en Braille



4 Observa el ejemplo en la *Figura 5*.Validen y depuren su programa para que cumpla las siguientes condiciones:

- Mostrar el mensaje si oprime el botón A.
- Mostrar el símbolo número (#) por 1 segundo, al inicio y final del mensaje.
- Mostrar cada consonante durante 3 segundos.
- Mostrar cada vocal respetando los siguientes tiempos:
    - Vocal a: 1 segundo
    - Vocal e: 2 segundos
    - Vocal i: 4 segundos
    - Vocales o, u: 5 segundos
- Mostrar un símbolo menos (-) para separar letras repetidas, si las hay en el mensaje.
- Si se oprime el botón B, se debe ver el mensaje escrito en letras. Así, quien descifra el mensaje, podrá validar su respuesta.

5 Cuando su programa esté listo y funcionando, sigan las instrucciones de su docente para desplazarse a mirar los dispositivos de otros grupos, copiar sus mensajes secretos y tratar de decodificarlos escribiéndolos nuevamente utilizando el alfabeto.

¿?

¿Lograron descifrar los mensajes secretos creados por otros grupos?

¿Pudieron otras personas descubrir el mensaje que ustedes codificaron?

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

Revisa de forma individual los aprendizajes de la sesión y determina el grado al que los alcanzaste.

1. ¿Puedes utilizar entradas y salidas booleanas en un programa en *MakeCode*?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes codificar mensajes representados de forma visual no alfabética?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

3. ¿Puedes crear, validar y depurar un programa para que transmita un mensaje visual según especificaciones dadas?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No



Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", devuélvete y busca en los contenidos de esta sesión el significado de las palabras: entrada, salida, booleana y codificar. Una vez hayas aclarado lo que estos términos significan, revisa las actividades que desarrollaste y discute con tu compañera o compañero de grupo la relación que tienen con las preguntas anteriores. Si, después de esto, todavía tienes dudas, acude a tu docente por apoyo adicional.

Ahora toma unos minutos para analizar lo que has aprendido en esta guía. Ya sabes cómo programar la tarjeta *micro:bit* para transmitir palabras, números, imágenes e incluso mensajes codificados por medio del panel de LEDs. También aprendiste cómo lograr que la tarjeta o su simulador, interactúen con las personas para mostrar resultados diferentes dependiendo, por ejemplo, de si presionan o no un botón. ¿Para qué crees que podrías utilizar todo esto que has aprendido?

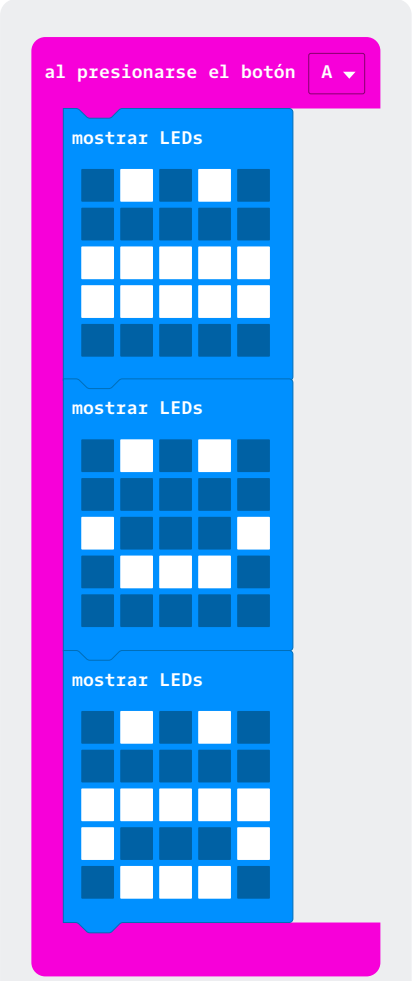
Si has trabajado con otros lenguajes de programación en bloques como *Scratch*, reflexiona un momento sobre las similitudes y las diferencias de este lenguaje y *MakeCode*.

¿Encontraste cosas en *MakeCode* que se parecen a *Scratch*?

¿Qué bloques encuentras en ambos lenguajes?

¿Cómo explicarías las diferencias?

Figura 6. Ejemplo de animómetro



Para ir más lejos

El siguiente reto adicional te permitirá probarte con lo aprendido hasta el momento. Vas a crear un programa que permita representar algunas emociones. Realiza una animación de íconos que te sirvan para ese fin.

¿Se te ocurre en qué situación podría ser útil contar con un programa como este?

Valida que tu programa cumpla con las siguientes condiciones:

- Al presionar el botón A, muestra una representación visual de la emoción de enojo.
- Al presionar el botón B, muestra una representación visual de la emoción de tristeza.
- Al presionar los botones A y B a la vez (A+B), muestra una representación visual de la emoción de asombro.

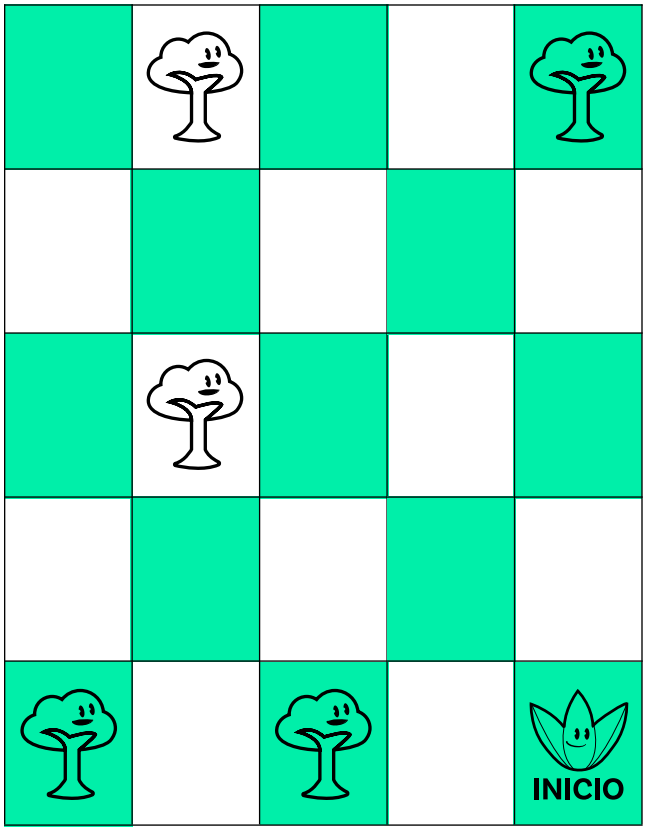
Mira el ejemplo que se presenta en la *Figura 6*, para inspirarte a crear tu "animómetro".

Actividad opcional para seguir consolidando aprendizajes

Ahora, toma unos minutos para analizar lo que has aprendido en esta guía. Ya sabes cómo programar la tarjeta *micro:bit* para transmitir palabras, números, imágenes e incluso mensajes codificados por medio del panel de LEDs. También aprendiste cómo lograr que la tarjeta o su simulador interactúen con las personas para mostrar resultados diferentes dependiendo, por ejemplo, de si presionan o no un botón. ¿Cómo podrías utilizar lo aprendido?

¿Qué tal si haces un dibujo que ilustre algunos usos de mensajes visuales programados?

Anexo 1.1 Tablero 1



Símbolo	Instrucción
▲	Tomar y levantar una ficha de la pila de inicio.
■	Bajar y soltar la ficha en la casilla actual.
→	Mover la ficha una casilla a la derecha.
←	Mover la ficha una casilla a la izquierda.
↑	Mover la ficha una casilla hacia el frente.
↓	Mover la ficha una casilla hacia atrás.

Pon a prueba tu programa

FICHA 1

FICHA 2

FICHA 3

FICHA 4

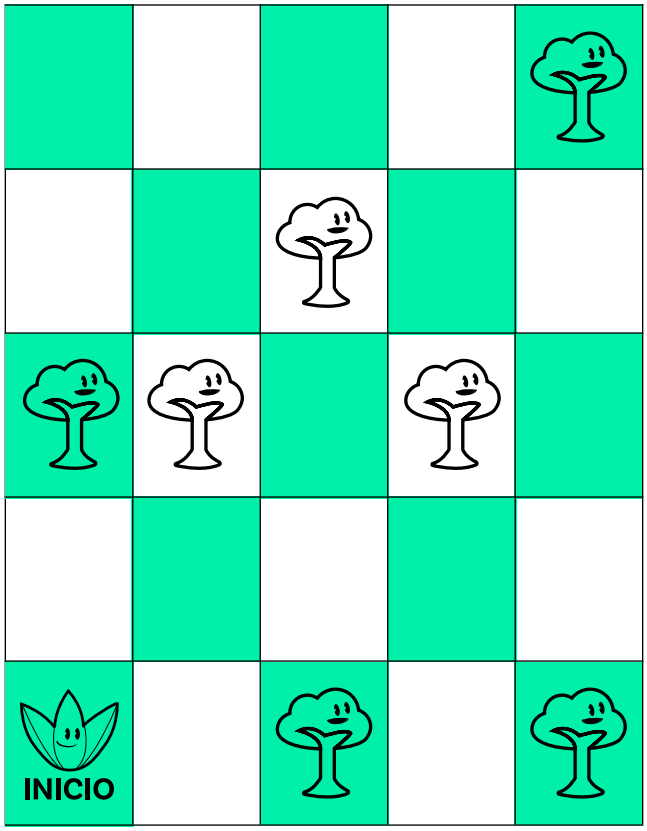
OBSERVACIONES

Procesador		Verificador
FICHA 1	☐	☐
FICHA 2	☐	☐
FICHA 3	☐	☐
FICHA 4	☐	☐

¿Hay error? SÍ ☐ NO ☐

¿Cuál es el error? Corrígelo

Anexo 1.2 Tablero 2



Símbolo	Instrucción
▲	Tomar y levantar una ficha de la pila de inicio.
■	Bajar y soltar la ficha en la casilla actual.
→	Mover la ficha una casilla a la derecha.
←	Mover la ficha una casilla a la izquierda.
↑	Mover la ficha una casilla hacia el frente.
↓	Mover la ficha una casilla hacia atrás.

Pon a prueba tu programa

FICHA 1

FICHA 2

FICHA 3

FICHA 4

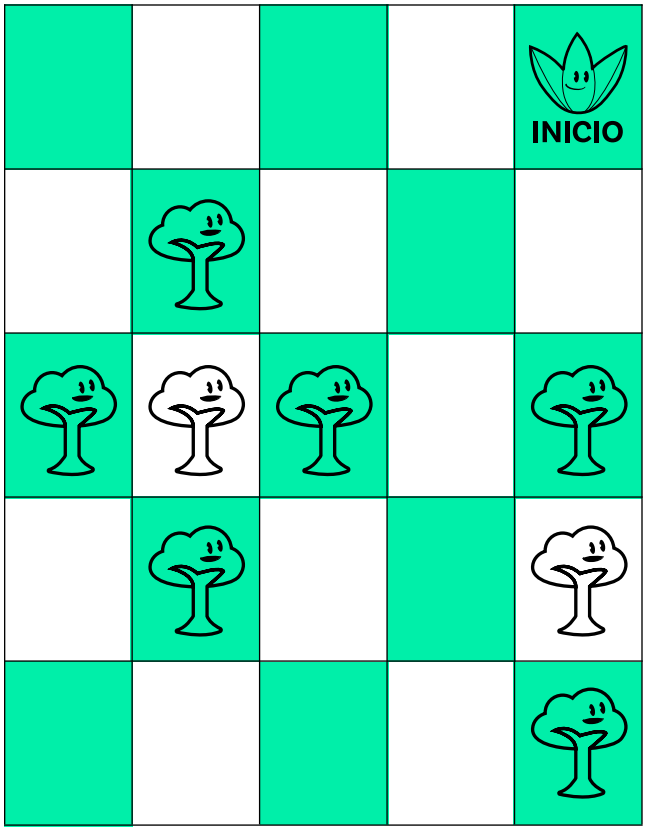
OBSERVACIONES

Procesador		Verificador
FICHA 1	☐	☐
FICHA 2	☐	☐
FICHA 3	☐	☐
FICHA 4	☐	☐

¿Hay error? SÍ ☐ NO ☐

¿Cuál es el error? Corrígelo

Anexo 1.3 Tablero 3



Símbolo	Instrucción
▲	Tomar y levantar una ficha de la pila de inicio.
■	Bajar y soltar la ficha en la casilla actual.
→	Mover la ficha una casilla a la derecha.
←	Mover la ficha una casilla a la izquierda.
↑	Mover la ficha una casilla hacia el frente.
↓	Mover la ficha una casilla hacia atrás.

Pon a prueba tu programa

FICHA 1

FICHA 2

FICHA 3

FICHA 4

OBSERVACIONES

Procesador		Verificador
FICHA 1	☐	☐
FICHA 2	☐	☐
FICHA 3	☐	☐
FICHA 4	☐	☐

¿Hay error? SÍ ☐ NO ☐

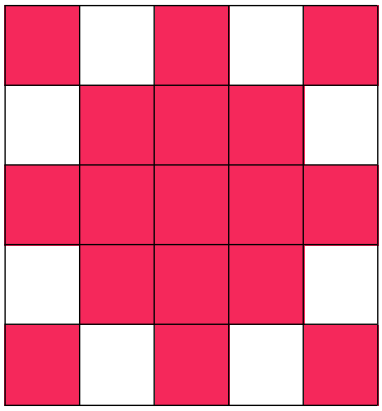
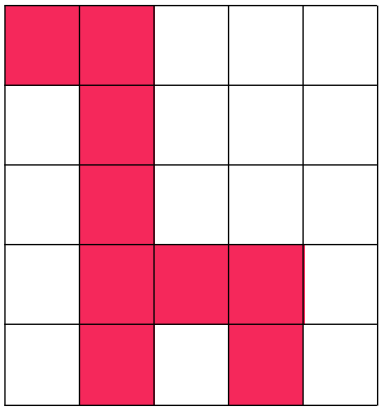
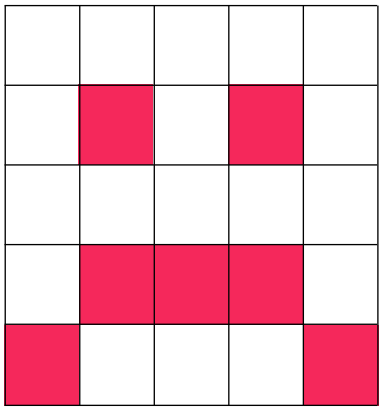
¿Cuál es el error? Corrígelo

Anexo 3.1 Planeación de una animación usando la matriz de LEDs

Nombres: _____

_____

1. El siguiente es el inicio de una historia contada con íconos y la ayuda de la *micro:bit*. Observen cómo las imágenes acompañan cada trocito de historia.

		
Una soleada mañana, Anita se fue a jugar al parque.	*Allí se encontró una pequeña jirafa de peluche en uno de los columpios.*	*Entonces, recordó que de camino al parque había visto a una mamá con su hijo llorando y señalando hacia el columpio...*

2. Ahora, siguiendo el ejemplo anterior, pueden escribir un breve resumen de una historia diferente que ustedes elijan o escribir una posible continuación del cuento de Anita y la jirafa de peluche.

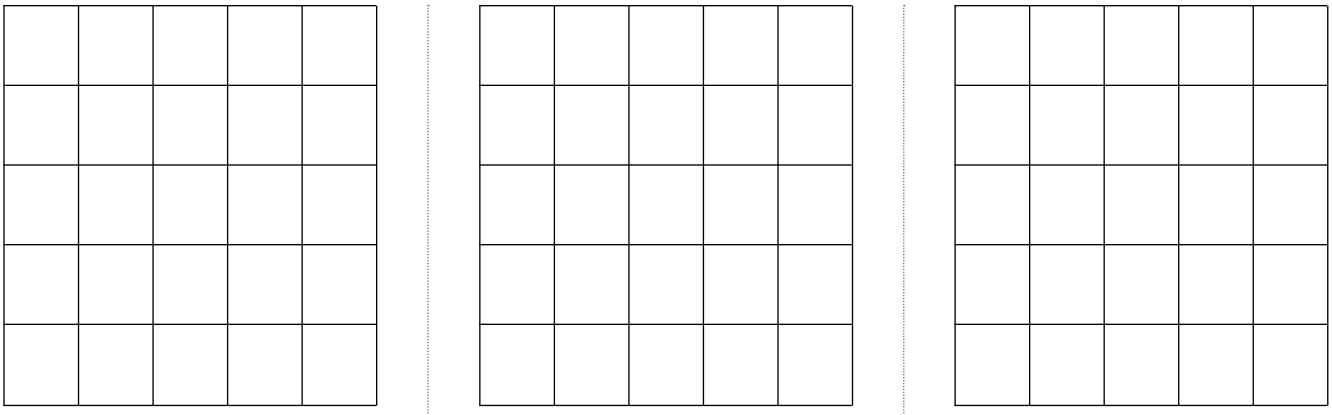
_____

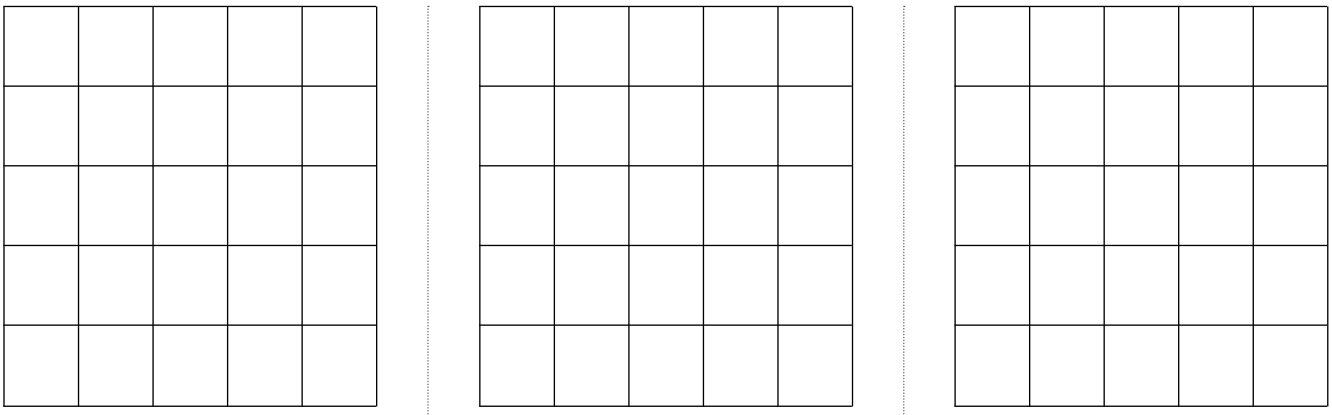
_____

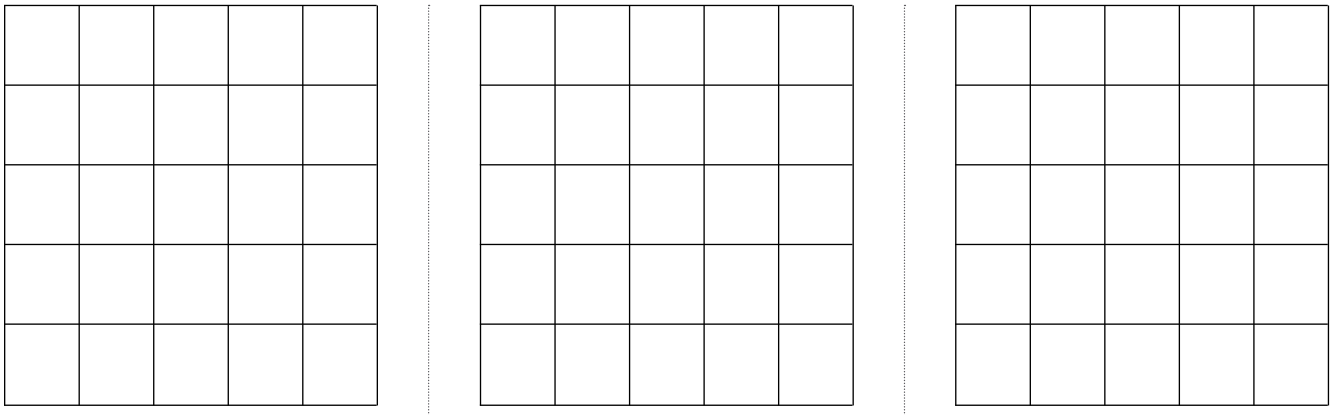
_____

_____

3 Ya que decidieron de qué se tratará su historia, elijan o diseñen los íconos que utilizarán para representar algunas de las escenas. Coloreen los LEDs de la *micro:bit* que van a programar para que se enciendan.







4. Después escriban las frases que acompañarán cada imagen.Piensen en las pausas que dejarán entre cada imagen. ¿Van a tardar todas el mismo tiempo o quieren que algún icono aparezca en la pantalla un poquito más que otro? Agreguen a la planeación anterior el tiempo que dejarán entre imágenes. Recuerden que 1 segundo son 1000 (ms).

5. ¡A programar! Regresen a *MakeCode* y programen la animación de acuerdo con la planeación que hicieron aquí.

6. Sigan las instrucciones de su docente para unirse a otra pareja de estudiantes y mostrarles la animación que ustedes hicieron mientras le cuentan su historia, y luego vean el trabajo de ese grupo.

TIC

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}