







Luces, códigos y mensajes visuales: Introducción a MakeCode







Estudiantes







MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Julián Molina Gómez Ministro TIC

Luis Eduardo Aguiar Delgadillo Viceministro (e) de Conectividad

Yeimi Carina Murcia Yela Viceministra de Transformación Digital

Óscar Alexander Ballen Cifuentes **Director (e) de Apropiación de TIC**

Alejandro Guzmán **Jefe de la Oficina Asesora de Prensa**

Equipo Técnico Lady Diana Mojica Bautista Cristhiam Fernando Jácome Jiménez Ricardo Cañón Moreno

Consultora experta Heidy Esperanza Gordillo Bogota

BRITISH COUNCIL

Felipe Villar Stein Director de país

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

Marianella Ortiz Montes Jefe de Colegios

David Vallejo Acuña Jefe de Implementación Colombia Programa

Equipo operativo

Juanita Camila Ruiz Díaz Bárbara De Castro Nieto Alexandra Ruiz Correa Dayra Maritza Paz Calderón Saúl F. Torres Óscar Daniel Barrios Díaz César Augusto Herrera Lozano Paula Álvarez Peña

Equipo técnico

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

Edición y coautoría versiones finales

Alejandro Espinal Duque Ana Lorena Molina Castro Vanesa Abad Rendón Raisa Marcela Ortiz Cardona

Edición Juanita Camila Ruiz Díaz Alexandra Ruiz Correa

British Computer Society – Consultoría internacional

Niel McLean Jefe de Educación

Julia Adamson **Directora Ejecutiva de Educación**

Claire Williams **Coordinadora de Alianzas**

Asociación 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

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 Ramos

PUNTOAPARTE EDITORES

Diseño, diagramación, ilustración, y revisión de estilo

Impreso por Panamericana Formas e Impresos S.A., Colombia

Material producido para Colombia Programa, en el marco del convenio 1247 de 2023 entre el Ministerio de Tecnologías de la Información y las Comunicaciones y el British Council

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

ⓒ (i) (S) CC BY-NC 4.0

"Esta guía corresponde a una versión preliminar en proceso de revisión y ajuste. La versión final actualizada estará disponible en formato digital y puede incluir modificaciones respecto a esta edición"

Prólogo

Estimados educadores, estudiantes y comunidad educativa:

En el Ministerio de Tecnologías de la Información y las Comunicaciones, creemos que la tecnología es una herramienta poderosa para incluir y transformar, mejorando la vida de todos los colombianos. Nos guia una visión de tecnología al servicio de la humanidad, ubicando siempre a las personas en el centro de la educación técnica.

Sabemos que no habrá progreso real si no garantizamos que los avances tecnológicos beneficien a todos, sin dejar a nadie atrás. Por eso, nos hemos propuesto una meta ambiciosa: formar a un millón de personas en habilidades que les permitan no solo adaptarse al futuro, sino construirlo con sus propias manos. Hoy damos un paso fundamental hacia este objetivo con la presentación de las guías de pensamiento computacional, un recurso diseñado para llevar a las aulas herramientas que fomenten la creatividad, el pensamiento crítico y la resolución de problemas.

Estas guías no son solo materiales educativos; son una invitación a imaginar, cuestionar y crear. En un mundo cada vez más impulsado por la inteligencia artificial, desarrollar habilidades como el pensamiento computacional se convierte en la base, en el primer acercamiento para que las y los ciudadanos aprendan a programar y solucionar problemas de forma lógica y estructurada.

Estas guías han sido diseñadas pensando en cada región del país, con actividades accesibles que se adaptan a diferentes contextos, incluyendo aquellos con limitaciones tecnológicas. Esta es una apuesta por la equidad, por cerrar las brechas y asegurar que nadie se quede atrás en la revolución digital. Quiero destacar, además, que son el resultado de un esfuerzo colectivo: más de 2.000 docentes colaboraron en su elaboración, compartiendo sus ideas y experiencias para que este material realmente se ajuste a las necesidades de nuestras aulas. Además, con el apoyo del British Council y su red de expertos internacionales, hemos integrado prácticas globales de excelencia adaptadas a nuestra realidad nacional.

Hoy presentamos un recurso innovador y de alta calidad, diseñado en línea con las orientaciones curriculares del Ministerio de Educación Nacional. Cada página de estas guias invita a transformar las aulas en espacios participativos, creativos y, sobre todo, en ambientes donde las y los estudiantes puedan desafiar estereotipos y explorar nuevas formas de pensar.

Trabajemos juntos para garantizar que cada estudiante, sin importar dónde se encuentre, tenga acceso a las herramientas necesarias para imaginar y construir un futuro en el que todos seamos protagonistas del cambio. Porque la tecnología debe ser un instrumento de justicia social, y estamos comprometidos a que las herramientas digitales ayuden a cerrar brechas sociales y económicas, garantizando oportunidades para todos.

Con estas guias, reafirmamos nuestro compromiso con la democratización de las tecnologías y el desarrollo rural, porque creemos en el potencial de cada región y en la capacidad de nuestras comunidades para liderar el cambio.

Julián Molina Gómez Ministro de Tecnologías de la Información y las Comunicaciones Gobierno de Colombia

Grado 5º Guía 1



Guía de íconos



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:



Ħ

</>

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 introduce 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 como los botones y salidas como la pantalla de un microprocesador como la *micro:bit* para recuperar o transmitir datos.

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.

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.







esion

Aprendizajes esperados

Al final de esta sesión se espera que puedas:



Reconocer en una experiencia de programación desconectada los conceptos de algoritmo, programa, procesador y persona que programa.



Proponer, ejecutar y verificar una secuencia de instrucciones para realizar una tarea.

Material para la clase

Anexos 1.1, 1.2, 1.3 0 \bigcirc 12 fichas







Duración sugerida



15%

15%

70%

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



Manos a la obra



Esta sección corresponde al 85% de

avance de la sesión

Desconectadas

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

La misión de su 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 fichas¹ representan las "cámaras" que se van a ubicar allí.

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

Anexos

Anexo 1.2



(1

2

Anexo 1.3

Arrow high grow mathematical set of the set of th	Anexa 13	labiero 3				Simbolo	Instruc	nolon
Image: Second secon							Tomary Toha de	levantar una la pita de inicio
Image: Section of the section of					M		Bijie yn Ia caella	oftar is fichs er activit
Image: Section 1 Image: Section 2 Image: Section 2					NICO	-	Mover la a la dere	i ficha una casi Icha.
Normal Action Normal Action Image: Action Image: Action <		P				+	Mover is a is loga	ificha una casi ierda
Image: Second	1	~	1		1	†	Mover's Pack of	ficha una casi fierte.
Common Company Processor Processor P	T	T	T		T	+	Mover's Pack at	fiche une cest
Project Project Project Project		P			P			2015
Pois purch to regime (2012) Pois purch to regime (2014)					100	Processol Erchant	×	Verificador
Pos. particle tympere (2014)					r	FICHA 2		
Per s prode its progenie 2018 1	_				-	FICHA 3 FICHA 4		
		ton a prueb	a ha progra	-		(Hay on a (Cubi et a	r si	NO
PO42		041						
2043	1	1042		TT.				
FD44.4	1	1048.3						
	ž	1048.4						
				±±				

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:

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 nel as 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.

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:

- 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.
- El robot no puede moverse por encima de las plantas ni de fichas que ya estén en el tablero.
- c. El programa se escribe utilizando los símbolosinstrucciones 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).

Símbolo	Instruccion
	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.
1	Mover la ficha una casilla hacia el frente.
ł	Mover la ficha una casilla hacia atrás.

Figura 1. Lenguaje de flechas para programar los desplazamientos

Ejemplo 1.

Posibles programas para mover las fichas



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

OE	OBSERVACIONES						
Procesador		Verificador					
FICHA 1							
FICHA 2	\checkmark						
FICHA 3	X						
FICHA 4							

3

4

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.

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



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.

 ¿Puedes reconocer en una experiencia de programación desconectada los conceptos de algoritmo, programa, procesador y persona que programa?

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.

Lee ahora el siguiente recuadro:



Los procesadores de los computadores, robots, lavadores, 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. 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

Al final de esta sesión se espera que puedas:



</>

</>>

Reconocer la interfaz del editor de bloques *MakeCode*.

Utilizar el editor *MakeCode* para crear un programa que muestre

números o íconos en pantalla.

Visualizar el código creado en el

simulador de la micro:bit.

Material para la clase

O Dispositivo con acceso a MαkeCode



Duración sugerida



15%

70%





Educación







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 *Figura 1*, al igual que en un rompecabezas, todo lo que has aprendido hasta el momento se conecta. Por ejemplo, una persona que programa 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!

;Hola!

Con esta guía empezarás a aprender el lenguaje de programación en bloques llamado *MakeCode*. Con él podrás crear programas que se ejecutan con un computador de bolsillo llamado tarjeta *micro:bit*, e incluso se pueden simular desde este lenguaje, si no cuentas con una tarjeta física.

Enlace

Puedes acceder al editor *MakeCode* en



También se puede acceder a una versión de MakeCode en una aplicación gratuita para el celular. En la Figura 2 se muestra una micro:bit. Obsérvala cuidadosamente.

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



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.

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)
 - 2 botones programables
 - D) Altavoz
 - 🗉 🔵 Logo tipo botón táctil
 -) Pantalla de LEDs

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 jovenes y ayudarles a usar sus habilidades de programación para resolver problemas.





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 enlace que se presenta en la *Figura 3*. Al hacerlo, verán un pantallazo como el siguiente.

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



Figura 9. Tour de reconocimiento de la plataforma



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 *Figurα 8*.

Figura 8. Selección lenguaje

Microsoft Omicro.bit		🗶 Diogues 📰 Javaticipt 🗠		#	<	0	٠	Iniciar sesión 🥷
	Seleccione un idioma			0				
·D D	English English	العربية Arabic	български Bulgarian					
	Català Catalan	Čeitina Czech	Cymraeg Welsh					
0 1 2 3V KHD	Darisk Darish	Deutsch German	Еλληνικό Greek					
	Español (España) Spanish (Spain)	Suomi Fineish	Français French	-				
	עברית Hebrew	Magyar Hurgarian	Íslanska Icelandic					
	Italiano Italian	日本語 Japanese	현국어 Korean					
Descarger e++	Nederlands	Norsk bokmål	Norsk nynorsk					~ • •

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 *Fígur*α 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.

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

2

3

4

al iniciar	
mostrar número	0

Figura 13. Código para mostrar un ícono



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



Ubiquen el bloque mostrar número 0 adentro 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.

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.

 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 *Figur*α 14.

¿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.

Grado 5º Guía 1

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.

- ¿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
-) ¿Puedes visualizar el código creado en el simulador de la micro:bit?
 - Sí Sí

3

-) 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. 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 _____

Finaliza completando los espacios en blanco en la siguiente gráfica. Agrégales imágenes que consideres adecuadas para ayudarte a recordar mejor.







esion **C**-

Aprendizajes esperados

Al final de esta sesión se espera que puedas:



Crear un programa que genere una animación sencilla en MakeCode basada en secuencias de íconos.

Duración sugerida



33%

34%



Aplicar la descomposición a la creación de animaciones con ayuda de la micro:bit.

Material para la clase

Acceso a MakeCode \bigcirc 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?

 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.

- 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.
- 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.

Code P Gala		Acces Encodered
Anexe 3.1 Planeación de una anim	ación usando la matriz de LEDs	
Nombre		
Nombre:		
Una solitada mañana, Anlia an fue a Jugar d'parque	All se encortó-una propueto julió de pelacher en uno de los columpios.	Entones, recentlo gui de comise ol pargue hobia vista a una mande can su higi à lonada gui alfanta



Glosario

s?

- 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 1.000. 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.

¿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.

Una vez tengas todo claro, ponte a prueba. 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 de vida de una mariposa? Sigue las instrucciones de tu docente para organizarte en grupos de 3 a 5 personas.

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 como esta:



Completen los espacios en blanco en el siguiente gráfico resumen:



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

4







Aprendizajes esperados

Al final de esta sesión se espera que puedas:





33%

34%

33%

Predecir los resultados de programas sencillos.

Reconocer cuando el resultado de un programa no es el esperado.



</>>

</>>

Depurar programas según un objetivo planteado.

Material para la clase

O Dispositivo con acceso a MakeCode









Sesión 4 Estudiantes

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 alguna animación en el simulador del *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 (\checkmark).

- Esperar 200 ms y mostrar el chulo completo.
- J Luego esperar 250 ms.

para siempre mostrar icon pausa (ms) 200 mostrar LEDs pausa (ms) 200 control de este.

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

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*.



Figura 3. Programa 1 para depurar

mostra	cadena	G14 G	SANTAN	DER 19M	(in'
pausa	(ms) 3000 ·	•			
mostra	cadena	07 30	AM"		
pausa	(ms) 2000 ·				
mostra	cadena 🖉	27/10	2024"		

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 organizarte en pareja 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.

E	scribe tu explicación aquí:
_	
_	
_	
P	rograma 2 (Figura 4)
P	rograma 2 (Figura 4) scribe tu explicación aquí:
- P E	rograma 2 (Figura 4) scribe tu explicación aquí:
- Е	rograma 2 (Figura 4) scribe tu explicación aquí:

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

Ahora sí, ingresen a *MαkeCode* 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	SI	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 Transporte² como la de una ruta de TransMilenio en la ciudad de Bogotá y el segundo mostraba información de un sistema de transporte intermunicipal³. 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.
- 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:

- 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.



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:

ŝ

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?

¿Cómo se sentirían si tuvieran que tomar uno de los buses de las rutas de transporte de los ejemplos y no pudieran identificar dónde tomarlo o que se fueran a subir y no pudieran saber hacia dónde va, por culpa de un error en la programación del mensaje?

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

Pistas:

	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.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

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 que aparecen en la página anterior, descubre las palabras secretas correspondientes y escríbelas en los espacios vacíos identificados por los números.

Figura 5. Gráfico de anclaje



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
O Parcialmente
O No
3 ¿Puedes depurar programas según un objetivo planteado?
🔘 Sí
O Parcialmente
O No
E. Citure versus stars forevers "Developments" o "A (or you" wive two
notas sobre la actividad de análisis de los 2 programas. Luego,

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.







Aprendizajes esperados

Al final de esta sesión se espera que puedas:



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.

Material para la clase

O 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 LED 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.

En programación a las entradas y salidas de información se les llama **booleanas**. Observa la *Figura 4*.

Figura 4. Botón como entrada y salida booleana

Al tener entradas de información como el botón A, pueden suceder dos cosas: oprimirlo o no oprimirlo. Si la indicación es que al oprimir el botón A se muestre la cara feliz, se obtiene un valor verdadero. En cambio, si no se oprime, no se ve la cara feliz, obteniendo un valor falso. De esta forma reconocerás que un **booleano** solo puede tener dos valores: verdadero o falso.

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 (*Figurα 6*) 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 6. Línea Braille

Figura 5. Alfabeto Braille

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.

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".

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

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. Observa el ejemplo en la Figura 7.

Figura 7. Uso del bloque mostrar LEDs para representar una letra en Braille

2

 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.
- Borrar la pantalla si se oprime el botón B. (Encontrarás este bloque en la caja de herramientas, entre los bloques básicos).

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 en español.

22

5

¿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.

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

- 🔵 Sí
 -) Parcialmente
 - Aún no

¿Puedes codificar mensajes representados de forma visual no alfabética?

🔵 Sí

2

2)

-) Parcialmente
-) No
- ¿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?

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

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 8. 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 8*, para inspirarte a crear tu "animómetro".

Anexo 1.1 Ta	ablero 1		
Ţ			INICIO

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.
1	Mover la ficha una casilla hacia el frente.
¥	Mover la ficha una casilla hacia atrás.

OBSERVACIONES

Procesador		Verificador
FICHA 1		
FICHA 2		
FICHA 3		
FICHA 4		
¿Hay error?	sí	NO

Pon a prueba tu programa

FICHA 1

FICHA 2

FICHA 3

FICHA 4

1					

¿Cuál es el error? Corrígelo

¿Hay error?

Grado 5º Guía 1

Anexo 1.2 ⊺	ablero 2				Sin
				(F)	
		(J			
(Ji)	Ŧ		Ţ		
INICIO		Ţ		Ţ	I

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.

OBSERVACIONES

Procesador	Verificador
FICHA 1	
FICHA 2	
FICHA 3	
FICHA 4	

NO

SÍ

¿Hay error?

Pon a prueba tu programa

FICHA 1

FICHA 2

FICHA 3

FICHA 4

Anexo 1.3 T	ablero 3			Símbolo	Instrucci	ón
					Tomar y lev ficha de la	vantar una pila de inicio.
					Bajar y solt la casilla ao	ar la ficha en ctual.
			INICIO	-	Mover la fi a la derech	cha una casilla 1a.
	(J1)			-	Mover la fi a la izquier	cha una casilla da.
	F				Mover la fi hacia el fre	cha una casilla ente.
	(ii)		(File)	♦	Mover la fi hacia atrás	cha una casilla 3.
	(1 ¹)					
	JE .				OBSERV	ACIONES
				Procesado	or	Verifi
			(11)	FICHA 1		
			JZ I	FICHA 2		
				FICHA 3		

FICHA 4

¿Hay error?

SÍ

¿Cuál es el error? Corrígelo

OBSERVACIONES Verificador

NO

Pon a prueba tu programa

FICHA 1

FICHA 2

FICHA 3

FICHA 4

- [
ł	 				
1					

	~	
	h	
-	_	

	Grado 5º	Guía 1	Anexos	Estudiantes
--	----------	--------	--------	-------------

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 peque en un	encor eña jiro o de lo	ntró u afa de os colu	na e peluc impios	he 3.		Entor cami visto hijo II	nces, r no al p a una	ecordo parque mam o y sei	ó que c e habíc á con : ñalanc	de x su do

(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 LED de la *micro:bit* que van a programar para que se enciendan. Después escriban las frases que acompañarán cada imagen.

Anexos **Estudiantes**

- 4 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 1.000 (ms).
- ⁵ ¡A programar! Regresen a *MαkeCode* 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.

