

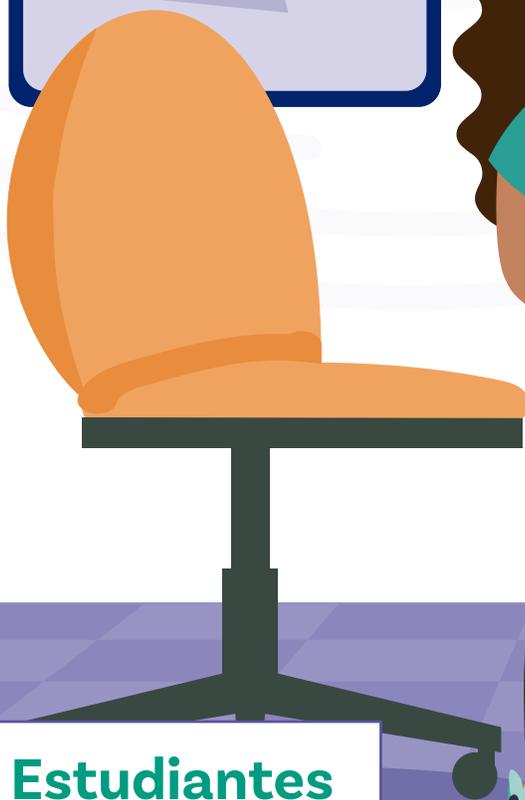
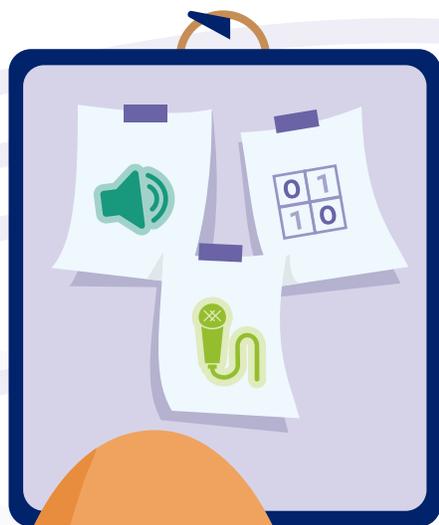
Arreglos que ayudan

Grado 7°

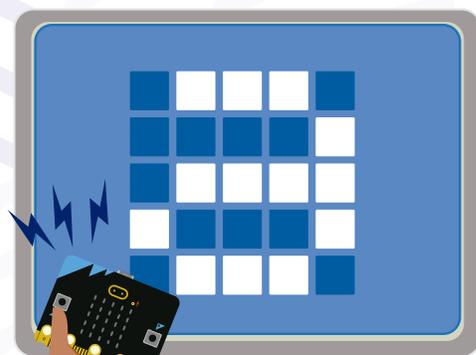
Guía 2



TIC



Estudiantes



Apoya:



Arreglos que ayudan

**Grado 7°**

**Guía 2**



**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/](https://creativecommons.org/licenses/by-nc/4.0/)

 **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 guía 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.

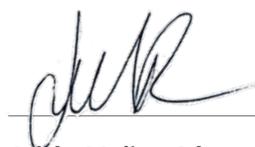
Estas guías han sido diseñadas pensando en cada región del país, con actividades accesibles que se adaptan a diferentes contextos, incluyendo aquellos con limitaciones tecnológicas. Esta es una apuesta por la equidad, por cerrar las brechas y asegurar que nadie se quede atrás en la revolución digital. 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 guías 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 guías, 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**



## Guía de íconos



Lógica,  
programación  
y depuración



Prácticas  
de datos



Computación  
física

## Aprendizajes de la guía

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



Usar variables para representar y almacenar datos y hacer operaciones sobre estas.



Definir el proceso de toma de datos y su almacenamiento en medios digitales para responder a preguntas, cuando sea posible.



Utilizar herramientas para depurar un programa.

## Resumen de la guía

Esta guía propone 5 sesiones de trabajo orientadas a aprender a usar arreglos para guardar datos tomados de sensores en el marco de computación física. El trabajo se hace usando el editor *MakeCode* y una actividad de computación desconectada.

## Resumen de las sesiones

## Sesión 1

Se explora el concepto de entrada y salida en un sistema digital y se utilizan las entradas y salidas de la *micro:bit*.

## Sesión 2

Se aprende a identificar y usar arreglos sencillos para guardar los datos y se presentan algunos ejemplos para poder registrar la información.

## Sesión 3

Se desarrolla una actividad desconectada que permite conocer una forma en la que se almacenan las imágenes en un computador.

## Sesión 4

Se utiliza uno de los sensores que tiene la *micro:bit* para tomar datos de una variable física a lo largo de un período y se almacenan estos valores en un arreglo. Los datos se utilizan para calcular la media.

## Aprendizajes de la guía



Utilizar los sensores integrados en dispositivos digitales como celulares y tarjetas con microprocesadores como la *micro:bit* y los puertos de entrada y salida de esta última para interactuar con el mundo.

## Sesión 5

Se hace un programa que permite reproducir parte de una melodía del interés de cada grupo.



## Conexión con otras áreas

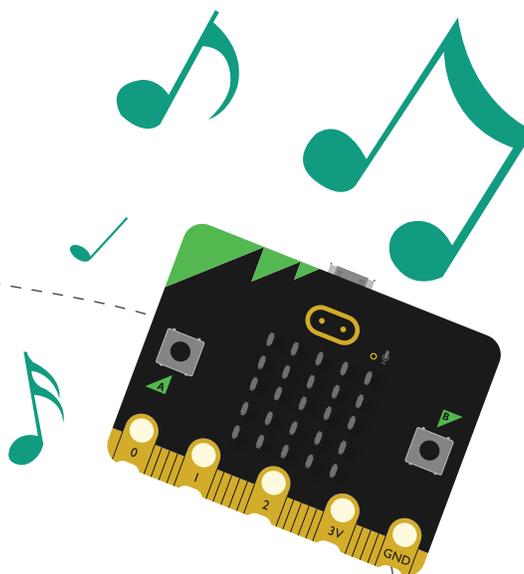
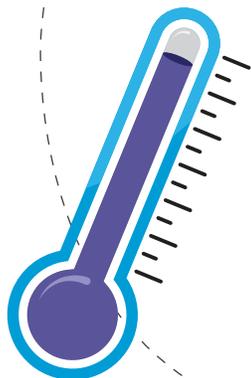
A continuación se presenta la conexión con otras áreas:

### Ciencias naturales

- La sesión 2 desarrolla actividades de computación física haciendo lecturas de variables físicas como la temperatura.

### Artes

- La sesión 5 permite reproducir melodías con el uso de *MakeCode* y el *micro:bit* lo que permitiría comprender y/o afianzar el conocimiento sobre las notas musicales.



# Sesión

# 1

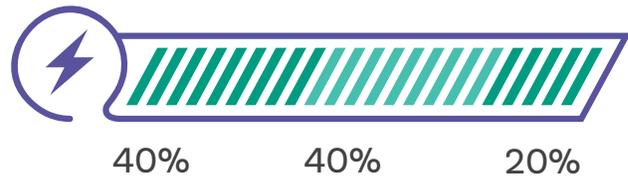
## Aprendizajes esperados

## Duración sugerida

Al final de esta sesión verifica que puedas:



Utilizar algunas de las entradas y salidas de un microprocesador como la *micro:bit* para interactuar con el entorno.



## Material para la clase

- Anexo 1.1.
- Anexo 1.2.
- Acceso a *MakeCode*.



Anexo

Anexo 1.1

El monitoreo de variables a menudo debe llevar a la toma de decisiones en función de la frecuencia con la que algo se presenta. Por ejemplo, no basta con que en un momento del día la temperatura sea un tanto elevada o un ruido sea muy alto, como para que se esta situación afecte a la salud.

Usualmente la duración de la situación anómala y el promedio de la variable medida son mejores indicadores de riesgo que una medida puntual.

En este reto te proponemos que diseñes un programa en MakeCode orientado a cuidar la salud de las personas expuestas frente a dos grandes riesgos: temperaturas elevadas y exceso de ruido.

Las especificaciones que debe cumplir son las siguientes:

- En el bloque de inicialización deben poderse parametrizar las siguientes variables:
  - Nivel mínimo de ruido seguro y porcentaje de mediciones altas permitidas.
  - Nivel máximo de temperatura segura y porcentaje de mediciones altas permitidas.
  - Tiempo entre toma de datos.
  - Número de mediciones a mantener.
- El programa deberá registrar la temperatura y el nivel de ruido cada intervalo definido en la inicialización.
- Mantener almacenadas las mediciones tanto de nivel de ruido como de temperatura en el número definido.
- Si alguna de las variables llega a estar por encima del valor de seguridad debe emitir una señal visual y sonora claramente identificable si es temperatura o ruido.
- Al apretar el botón A se debe entregar el porcentaje de mediciones sobre el valor límite en temperatura.
- Al apretar el botón B se debe entregar el porcentaje de mediciones sobre el valor límite en ruido.

El programa desarrollado deberá probarse en simulación usando MakeCode.

Lo que sabemos,  
lo que debemos saber



Esta sección corresponde al 40% de avance de la sesión

Empieza leyendo el reto que te proponemos en esta guía, el cual está en el Anexo 1.1. A lo largo de la guía, aprenderás lo necesario para resolverlo.

En esta sesión comenzarás a aprender sobre las entradas y salidas de un sistema digital.

Empecemos con esta primera pregunta relacionada con el concepto de entrada:

*¿Por cuáles medios las personas recogemos información del entorno?*

Discute con las otras personas sobre esta pregunta e identifica los sistemas de nuestro cuerpo que nos permiten recoger información del medio.

Para identificar alguno de nuestros sentidos, toma objetos con los ojos vendados o cerrados, tocando diferentes texturas o adivinando sonidos reproducidos. Realizar estas acciones te permitirá identificar, que por medio del tacto o de lo que escuchamos, podemos recolectar información y saber qué objetos estamos sintiendo o qué estamos escuchando.

Probablemente llegues a darte cuenta de que son justamente los 5 sentidos lo que nos permiten recoger información sobre lo que escuchamos, vemos, sentimos, olemos o saboreamos. Un sentido menos conocido es el de la propiocepción que nos permite tener conciencia corporal y del movimiento.

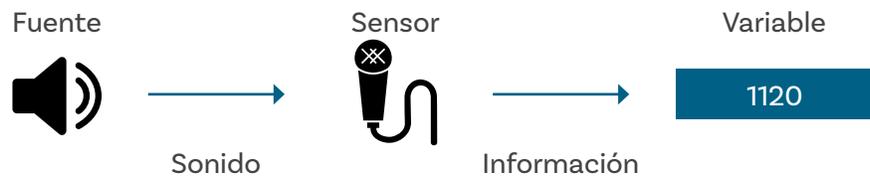
**Nota**

En la Guía 3 del grado 6 se abordan las primeras actividades de sensores y computación física en caso de que se requiera comprender este tema de manera inicial.

Nos llega tanta información a través de los sentidos, que nuestro cerebro debe seleccionar a cuál de toda esta información prestar atención y cuál de ella en cuenta. Este es un proceso en buena medida inconsciente. Por ello, a la mayor parte de la información que recibimos por nuestros sentidos, no le prestamos atención y de hecho somos incapaces de recordarla dado que nunca fue tomada en cuenta. Una cosa es, por ejemplo, los sonidos que nos llegan al oído y otra cosa es lo que realmente reconocemos, la información que seleccionamos.

En un ordenador, la situación es similar, como veremos. Podemos tener múltiples **sensores**, pero sólo cuando nuestro programa los interroga, la información se considera y se retiene.

Un sensor de la *micro:bit*, por ejemplo, el micrófono, está recibiendo sonido permanentemente, pero solo cuando el programa le pregunta al sensor por el sonido que capta y lo guardamos en una variable, tenemos información que puede ser útil. Examina el siguiente diagrama:

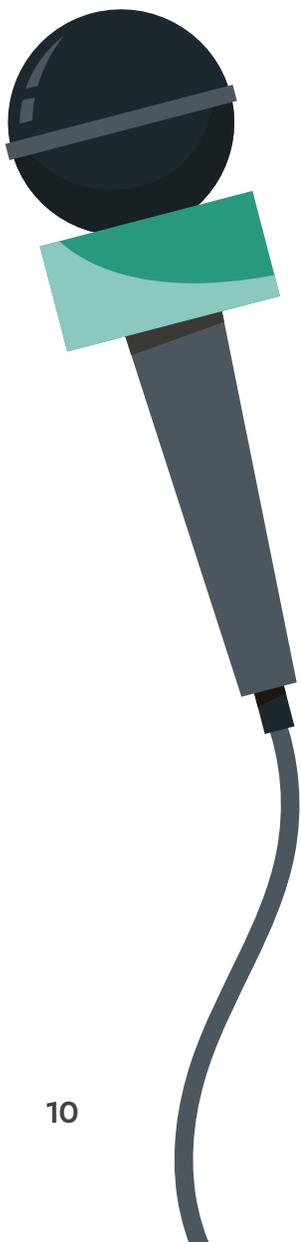


Usualmente cuando en un programa le pedimos a un sensor medir algo, nos da uno o varios datos de forma numérica. Es importante diferenciar el sensor de la información que nos da: un termómetro, el sensor, nos da la temperatura, la información.

Una entrada es el medio por el cuál un **dispositivo digital**, como un celular o un computador recibe información del exterior. Al igual que los seres humanos que tenemos 5 sentidos que son nuestras “entradas” de información, los dispositivos pueden tener diferentes entradas para recibir diferentes tipos de información.

En un celular la entrada es, por ejemplo:

- La entrada táctil de la pantalla
- El micrófono
- La cámara



Los celulares tienen otras entradas, como:

- El GPS que le permite saber dónde está.
- Sensor de movimiento que incluso puede contar pasos.
- Sensor de temperatura que le permite vigilar que la temperatura alta no dañe el equipo.
- Sensor de campo magnético que le permite saber dónde es el norte magnético en la Tierra.
- Sensor de nivel de iluminación, que le permite ajustar el nivel de iluminación de la pantalla para ahorrar energía.

Ahora piensa con tus colegas sobre la siguiente pregunta:



*¿Cuáles son las entradas de un computador?  
¿Cuáles con las entradas de un ascensor?*

Pasemos ahora al concepto de salida.

Piensa de nuevo en el cuerpo humano y responde a las siguientes preguntas:



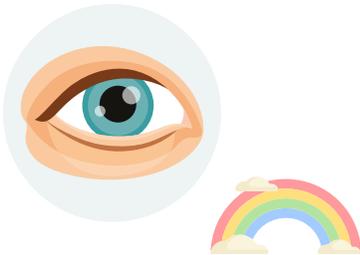
*¿Cómo nuestro cuerpo percibe las condiciones del ambiente como el clima, el ruido o los colores?  
¿Cómo nuestro cuerpo recibe y procesa esta información?  
¿Cómo realizamos acciones de acuerdo a las condiciones que percibimos?*

Probablemente identifiquen procesos como hablar, hacer señas, usar nuestros músculos para llevar a cabo acciones o usar lenguaje corporal y gestos como indicar NO con un dedo o la cabeza.

Pasemos a un sistema digital. Piensen en un celular:



*¿Cuáles pueden ser las salidas?*



Seguramente identificaste:

- La pantalla para mostrar cosas.
- El parlante para hacer sonar música o escucha a la persona con quien hablamos o indicar que hay una llamada.
- El vibrador para indicar algo cuando lo tenemos en silencio.

Piensa ahora en otros sistemas digitales:

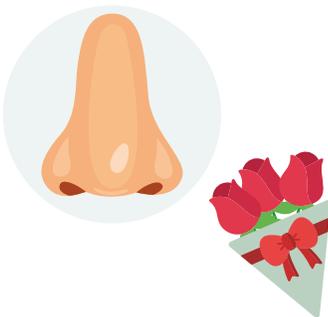


*¿Cuáles son las salidas de un computador?  
¿Cuáles son las salidas de un ascensor?*

Todos los sistemas digitales tienen entradas que les permiten recoger información del entorno y salidas que le permiten interactuar con el medio, incluso realizando acciones.

Examina ahora la *Figura 1*. Los robots en general no son como los muestran y rara vez se parecen a un humano.

Reflexiona con otras personas en el salón sobre cuales podrían ser las entradas y las salidas en este robot.



*¿Qué debe medir del entorno para hacer bien su labor? ¿Qué salidas tiene la computadora que maneja para poder realizar la tarea de pintar un vehículo?*

Y recuerda, los robots tienen muchas formas diferentes, usualmente diferentes a la forma humana. Su forma depende de su tarea.



**Figura 1.** Diferentes tipos de robots



### Glosario

- 
**Sensor:** un elemento que puede medir una variable física, como por ejemplo la temperatura o el nivel de iluminación.
- 
**Sistema digital:** hace referencia a artefactos que reciben, procesan e interactúan con el medio a partir de los resultados. Digital viene del hecho de que trabaja con información numérica, usualmente en base binaria donde todo se representa usando ceros y unos, gráficas, números, fotos o videos.

### Manos a la obra

### Conectadas



Esta sección corresponde al 80% de avance de la sesión

### Anexos

#### Anexo 1.2

Instrucción	Explicación	¿Es una entrada?	¿Es una salida?
	Muestra un corazón en la pantalla de LED de la micro:bit.		
	Realiza una acción si el botón A se oprime.		
	Realiza una acción si la tecla B se mueve.		
	Hace sonar un tono en dicho modo.		
	Reproduce una melodía a partir de datos.		
	Enciende un LED particular.		
	Borra la pantalla.		
	Dato de dirección de la brújula en la micro:bit.		
	Nivel de luz medido por la micro:bit.		

Organízate en grupos siguiendo las indicaciones de tu docente.

Es el momento de regresar al editor de *MakeCode* para trabajar con las entradas y las salidas de la *micro:bit*.

Deben explorar los componentes del menú y decidir cuáles son entradas y qué miden o registran y cuáles son salidas y cómo se supone que se comunican con el exterior.

Usen el Anexo 1.2 para identificar en cada caso si la instrucción se refiere a una entrada o a una salida de la *micro:bit*.

Una vez completen la tabla es hora de conectarnos al editor de *MakeCode*.

Revisen el siguiente programa en bloques. ¿Qué hace?

**Figura 2.** Programa en MakeCode



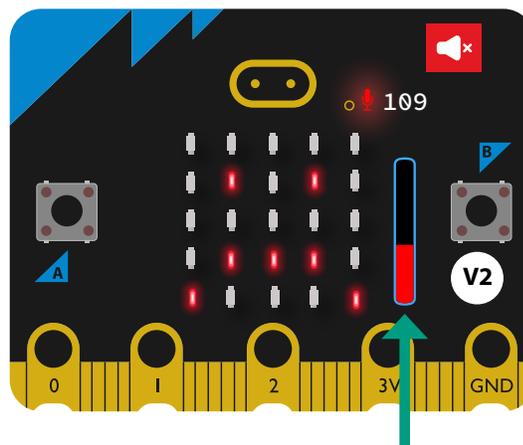
Tengan en cuenta que, en el editor, realmente no se mide la variable, sino que se puede cambiar de forma manual con el cursor. En la *Figura 3* aparece la barra para simular un nivel de sonido entrante.

Si tienen una *micro:bit*, podrán transferir el programa y probar con datos reales este pequeño programa.

Han trabajado con tres entradas (botones A y B, micrófono) y una salida (la pantalla de LED).

Si tienen tiempo, modifiquen el programa para medir otras variables (moverse, temperatura, girar, etc.) y alertar o cambiar la alerta a un pito.

**Figura 3.** Indicador de sonido entrante en simulador de MakeCode



## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

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

- 1 ¿Puedes utilizar algunas de las entradas y salidas de un microprocesador como la *micro:bit* para interactuar con el entorno?
- Sí
- Parcialmente
- Aún no

**Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas en *MakeCode*. 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.**

Recuerda que:

*Entradas en un sistema digital pueden ser muchas, como el teclado mismo o el ratón o una tableta para pintar.*

*Las salidas pueden ser incluso una impresora de papel, una impresora 3D o el accionamiento de un motor.*

Responde primero tú y luego comparte con tus compañeras(os), la siguiente pregunta:



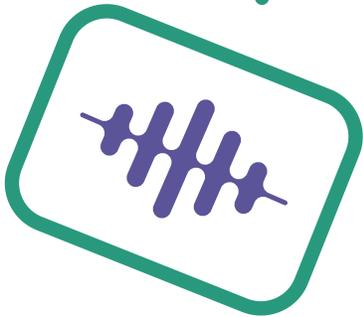
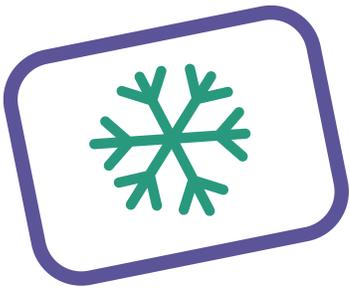
En el reto, ¿en qué te servirá lo que has aprendido?

Ahora te proponemos completar el siguiente esquema con tus aprendizajes:

Los sistemas computacionales tienen entradas y salidas

Entradas de la *micro:bit* como

Salidas de la *micro:bit* como



# Sesión 2

## Aprendizajes esperados

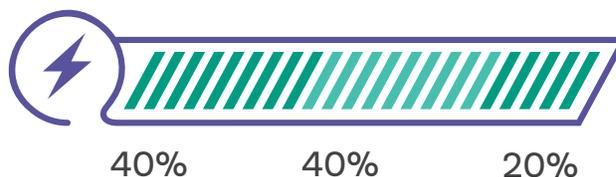


Utilizar arreglos para guardar y realizar operaciones sobre su contenido.



Utilizar bloques de repetición utilizando un índice que va incrementando en cada iteración.

## Duración sugerida



## Material para la clase

- Acceso a *MakeCode*.



**Nota**

Al referirnos a arreglos se usa la notación fila x columna, por ejemplo, 4x3, son 4 filas y 3 columnas.

**Lo que sabemos,****lo que debemos saber**

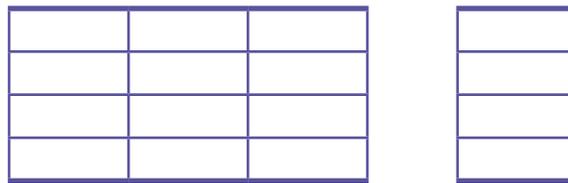
Esta sección corresponde al 40% de avance de la sesión

En la sesión pasada aprendiste o perfeccionaste el uso de entradas y salidas disponibles en la *micro:bit*.

Cuando usamos las entradas de la *micro:bit*, a menudo, tenemos que guardar no solo un valor sino una serie de valores. Esto sucede también para muchas otras tareas que hay que hacer en un computador.

Cuando se desean guardar varios datos lo podemos hacer en variables independientes, pero si, por ejemplo, el dato es de temperatura o iluminación, podríamos guardar datos cada cierto tiempo y mantenerlos para luego hacer algo con ellos.

Las estructuras en que se guardan los datos en un computador son múltiples y entre estas estructuras están los arreglos. Mira estos dos ejemplos:



A la izquierda se tiene un arreglo de dos dimensiones (de 4x3), mientras a la derecha se tiene un arreglo de una sola dimensión (4x1). A estos arreglos también se les llama **matriz**.

En esta sesión trabajarás con un arreglo de una sola dimensión.

Si, por ejemplo, tuviésemos que guardar 4 datos una opción es crear 4 variables. Recuerda cómo creamos estas variables. Estas podrían ser:

---

 Temperatura\_1

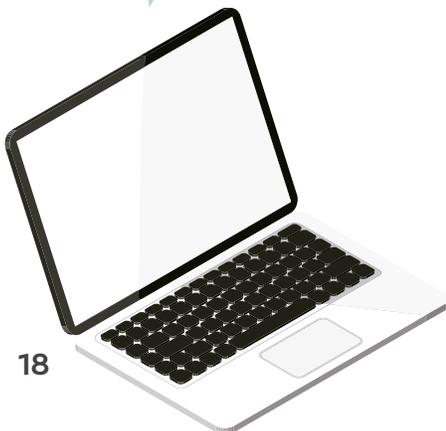
---

 Temperatura\_2

---

 Temperatura\_3

---

 Temperatura\_4


**Nota**

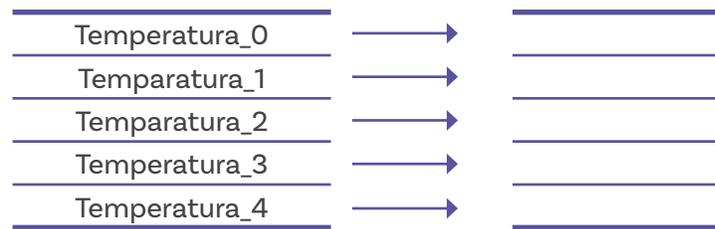
Es posible tener programas independientes que se ejecutan en paralelo como en este caso, donde hay un programa definido que se ejecuta cuando se oprime A y otro cuando se oprime B.

Sin embargo, los computadores no pueden hacer sino una tarea en un momento dado, de modo que si oprimes B antes de que termine A se pueden obtener resultados inesperados dado que la matriz no tendrá valores guardados en algunas de sus posiciones.

Sin embargo, este manejo lleva a programas de computador muy largos, dado que hay que hacer tener instrucciones para cada variable.

En un arreglo es importante poderse referir a cada variable dentro del arreglo.

Si se crea, por ejemplo, un arreglo llamado Temperatura de 4x1 lo que se tendría sería:



Cada lugar en esta variable que tiene varias posiciones se puede referenciar de forma independiente usando un “índice”. De modo que podemos referirnos directamente a cada “campo” en este arreglo usando un índice. Como el índice es un valor numérico, será entonces posible hacer un programa genérico y así solo cambiar el valor del índice para referirse a un campo o posición diferente en el arreglo.

Mira el siguiente ejemplo de lo que está almacenado en una variable tipo arreglo llamada Temperatura:

25,7
30,2
28,7
20,1
39,2

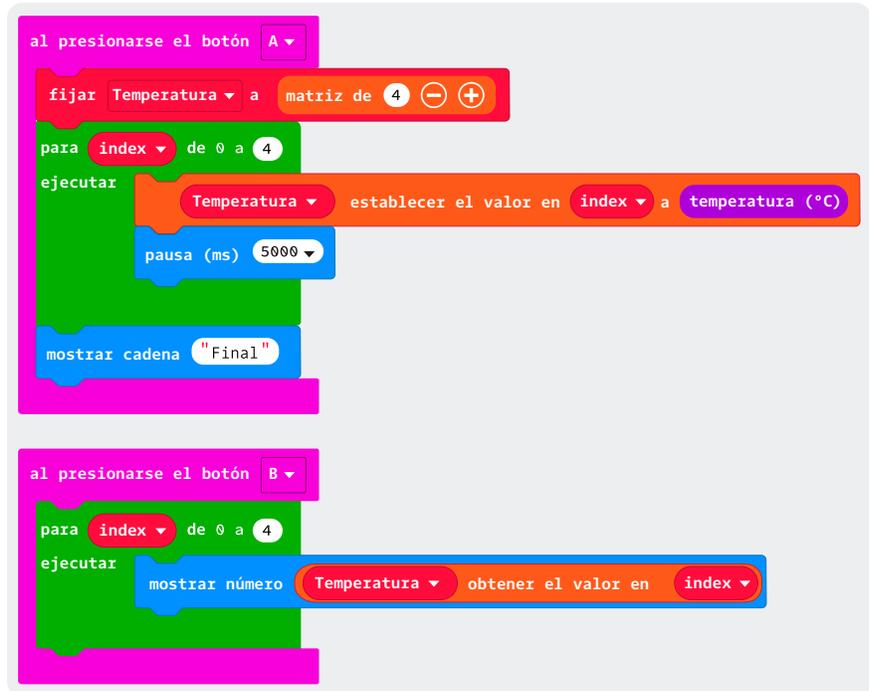
*Si, por ejemplo, indicamos Temperatura (3), ¿cuál es el valor al que nos referimos?*

*Si tenemos una variable **ÍNDICE** y esta tiene un valor de 2, ¿cuál será el valor de Temperatura (**ÍNDICE**)?*

Si contestaste a la primera pregunta que el valor es 20,1 y a la segunda pregunta 28,7 estás en lo correcto y has comprendido el manejo de este tipo de arreglos.

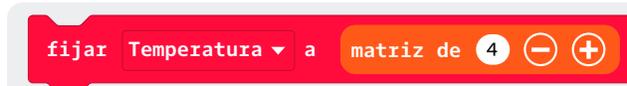
Es hora de examinar este tema en la programación en bloques de MakeCode.

**Figura 1.** Programación de matriz en MakeCode



Analiza el programa cuando se presiona el botón A. Verás varios elementos en él:

- Utiliza comandos del menú en la sección de arreglos:



- Crea una matriz de 4x1, esto es de 5 filas dado que van del elemento 0 al 4.
- Trabaja con un bloque de repetición verde con una variable llamada `index` que va cambiando de 0 a 4:



- Se usa esta variable `índex` para indicar la posición en la matriz en que se guarda el dato que se lee de temperatura:



Temperatura > establecer el valor en índex > a temperatura (°C)

- Se espera 5 segundos antes de avanzar `índex` en 1, tomar un nuevo dato y guardarlo en una nueva posición de la matriz.
- Se termina cuando `índex` llega a 4 y se muestra el resultado.

Ahora es tu turno de analizar la parte del programa relacionada con oprimir el botón B.

## Glosario



**Matriz:** es una estructura de datos organizada en filas y columnas que se utiliza para representar y manipular conjuntos de números, elementos o valores.

## Manos a la obra

### Conectadas

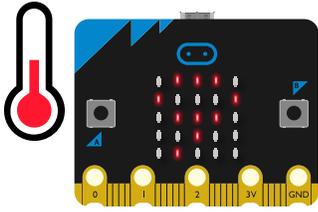


Esta sección corresponde al 80% de avance de la sesión

Organízate en grupos de dos o tres personas siguiendo las indicaciones del docente.

El objetivo principal es crear un programa que registre y analice datos de temperatura, lo que te permitirá comprender cómo automatizar tareas y procesar datos de manera eficiente, para utilizar el conocimiento y las habilidades sobre el uso de sensores.

El programa debe registrar la temperatura cada 10 segundos durante un período de 2 minutos. Se inicia esta instrucción cuando se presiona el botón A.



Además, deberá calcular y mostrar el promedio de las temperaturas registradas al presionar el botón B.

Mientras trabajan en esta actividad, asegúrense de dividir las tareas dentro del equipo para aprovechar las fortalezas de cada integrante. Por ejemplo, pueden asignarse roles como la programación del registro de datos, la implementación del cálculo del promedio o la verificación y pruebas del código.

Reflexionen sobre la utilidad de esta actividad para abordar el reto en el Anexo 1.1.

No duden en experimentar o buscar alternativas para mejorar el programa.

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes esperados de forma individual respondiendo las preguntas de forma que mejor reflejen tu progreso:

- 1 ¿Puedes utilizar arreglos para guardar y realizar operaciones sobre su contenido?
  - Sí
  - Parcialmente
  - Aún no
- 2 ¿Puedes utilizar bloques de repetición utilizando un índice que va incrementando en cada iteración?
  - Sí
  - Parcialmente
  - Aún no

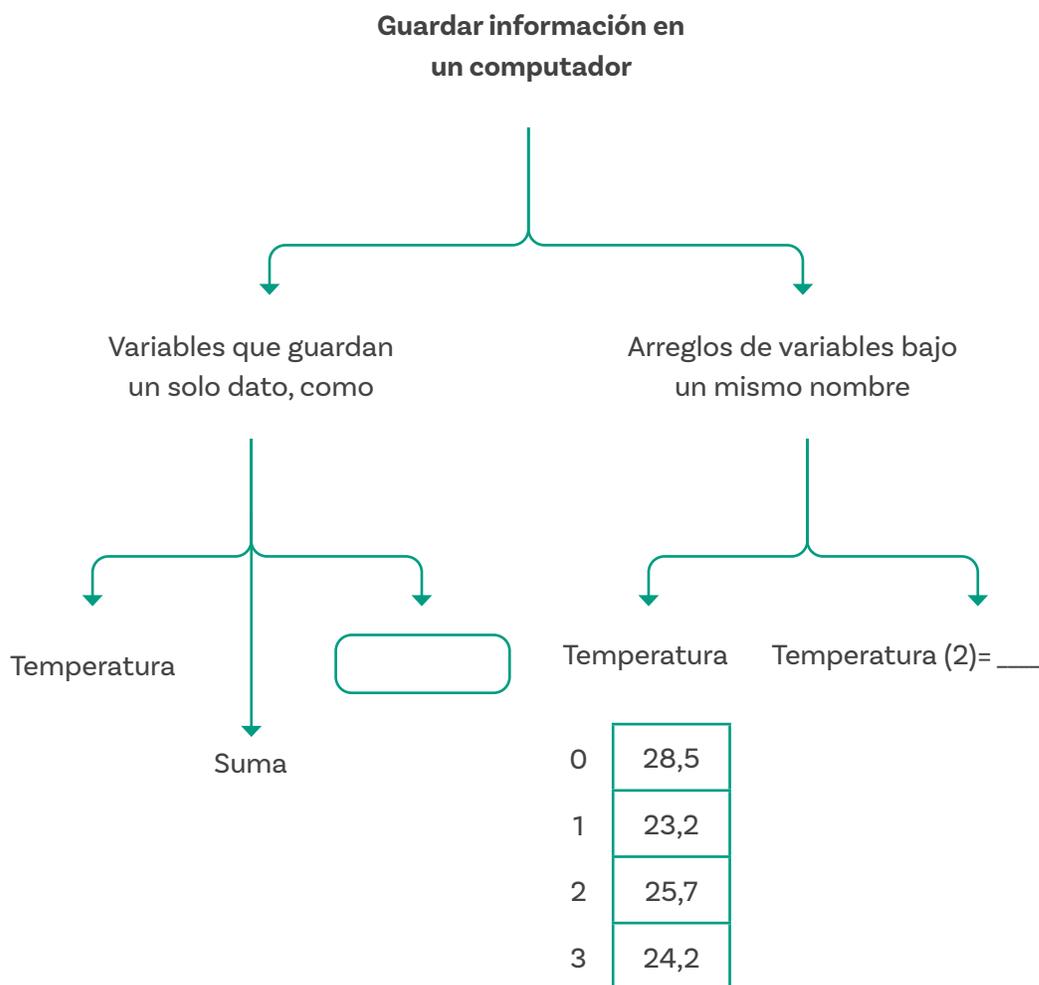


**Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas en *MakeCode*. Luego discute con tus compañeras y compañeros de grupo lo que se hizo en cada momento de la actividad y el rol al que correspondía. Si todavía te quedan dudas, consúltale a tu docente.**

Responde primero tú y luego comparte con tus compañeras(os), la siguiente pregunta:

*En el reto, ¿en qué te servirá lo que has aprendido?*

Al final de cada sesión se espera que hagas un gráfico o esquema que resuma lo aprendido. Te sugerimos completar el que te proponemos:

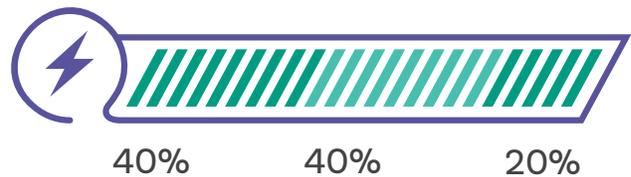


# Sesión 3

## Aprendizajes esperados

## Duración sugerida

Al final de esta sesión verifica que puedas:



Utilizar arreglos para guardar imágenes.



Explicar formas de codificar imágenes.



Realizar cálculos a partir de imágenes en píxeles.

## Material para la clase

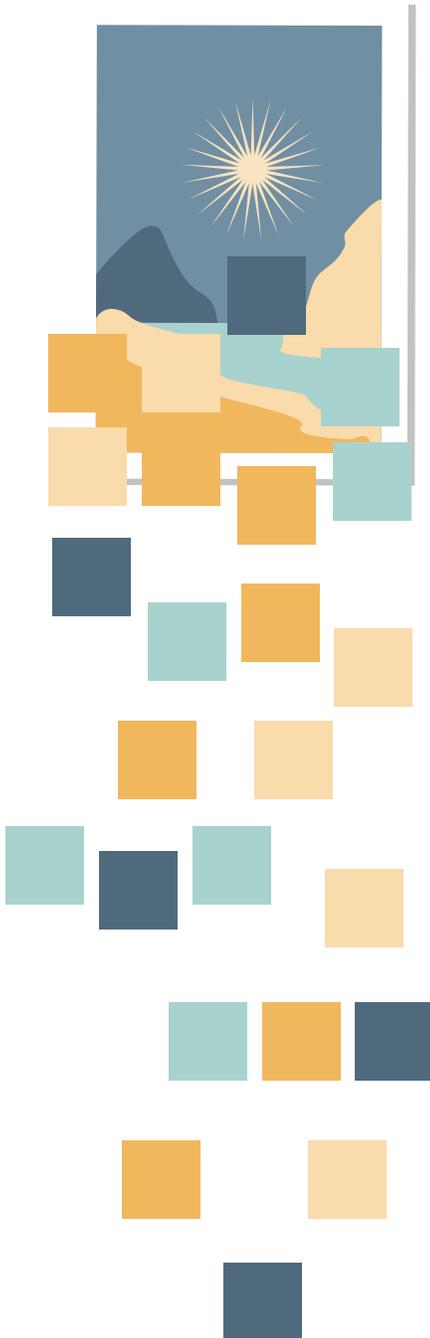
- Anexo 3.1.
- Anexo 3.2.
- Hoja de papel cuadriculado.
- Opcional:** Acceso a *MakeCode*.



## Lo que sabemos, lo que debemos saber



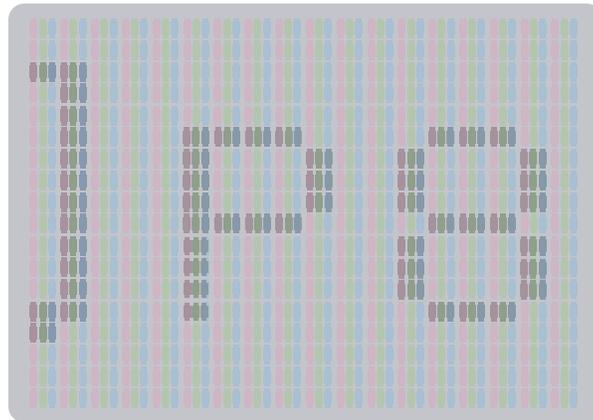
Esta sección corresponde al 40% de avance de la sesión



Uno de los usos de arreglos de dos dimensiones es guardar imágenes en un computador. Un computador, solo trabaja con números en formato binario (ceros y unos), de modo que imágenes, texto, voces, y música se convierten en datos.

En esta sesión trabajarás con alternativas para guardar imágenes en un computador usando arreglos.

Una imagen, para poderse guardar en un computador, se fracciona en pequeñas partes llamadas píxeles. Examina la siguiente imagen de un pedazo de pantalla de computador ampliificada:



De hecho, para que no la veamos así, el número de píxeles de una pantalla moderna es muy alto. Por ejemplo, en el formato que se llama FULL HD se manejan 1920x1080 píxeles lo que equivale a tener más de 2 millones de píxeles en una sola imagen. Para una película, por ejemplo, se presentan entre 30 y 60 imágenes por minuto lo cual implica que se requieren más de 60 millones de datos por minuto con el fin de reconstruir la imagen.

Pero veamos ahora cómo guardar una imagen en un arreglo. Existen diferentes formas de codificar una imagen. Una de ellas es usar un arreglo del mismo tamaño que el número de píxeles de la imagen y guardar en cada campo un valor que dice, por ejemplo, si el píxel está negro (un uno) o blanco (un cero).

Examina el siguiente arreglo con la información de una imagen:

0	1	1	1	0
0	0	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	1

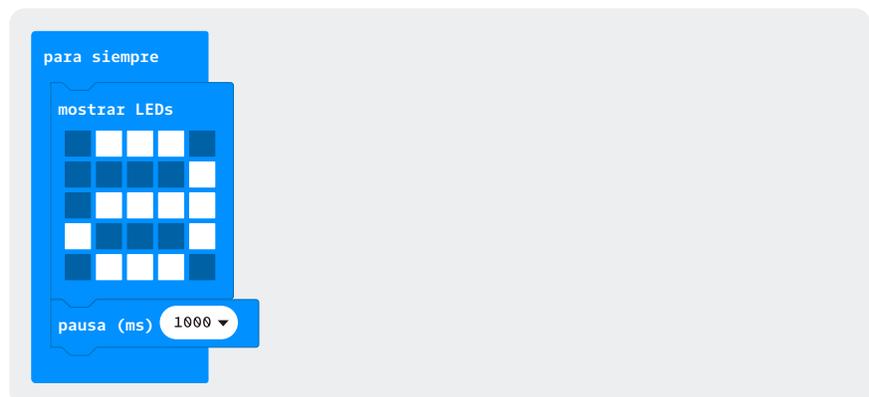
Es posible que si prestas mucha atención puedas detectar de qué se trata, particularmente si la observas desde lejos para que tu cerebro logre ver el conjunto y no cada elemento.

Ahora te sugerimos usar el siguiente arreglo para pintar en negro los cuadros donde hay un 1 y dejar en blanco los cuadros donde hay un 0.


Seguramente será más fácil observar de qué letra se trata.

Si tienes tiempo, usando el editor de *MakeCode*, te invitamos a mostrar en pantalla, separadas por un segundo, cada una de las vocales, en un programa que se repite indefinidamente.

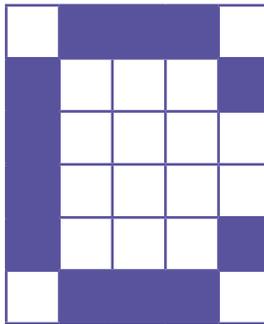
A continuación, un ejemplo que muestra la primera vocal:



Esta forma de almacenar datos no necesariamente es la más efectiva. Si son pocos los puntos en negro, se tendría una matriz con muchísimos blancos.

Una forma de ahorrar memoria en el computador es usar una codificación mejor.

Este es un posible ejemplo para la letra c:



1, 3, 1
0, 1, 3, 1
0, 1, 4
0, 1, 4
0, 1, 3, 1
1, 3, 1

- Se procede línea a línea.
- Se van indicando alternadamente cuántos pixeles blancos, cuántos negros, cuántos blancos, y así hasta terminar.
- Cada línea debe sumar el número de pixeles en cada línea, en este caso 5.

Algunas de estas formas de almacenar la información pueden reducir la cantidad de datos a guardar.



## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes esperados de forma individual respondiendo las preguntas de forma que mejor reflejen tu progreso:

- 1 ¿Puedes utilizar arreglos para guardar imágenes?
  - Sí
  - Parcialmente
  - Aún no
  
- 2 ¿Puedes explicar formas de codificar imágenes?
  - Sí
  - Parcialmente
  - Aún no
  
- 3 ¿Puedes realizar cálculos a partir de imágenes en pixeles?
  - Sí
  - Parcialmente
  - Aún no

**Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas. Luego discute con tus compañeras y compañeros de grupo 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.**

Resume lo aprendido en un esquema o diagrama del mismo tipo de los que viste en las sesiones previas.

# Sesión 4

## Aprendizajes esperados

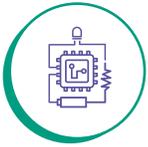
Al final de esta sesión verifica que puedas:



Utilizar sensores para capturar información.



Organizar datos en arreglos.

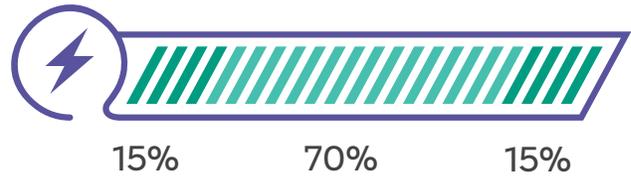


Utilizar la información en un arreglo para realizar cálculos.



Utilizar herramientas de depuración.

## Duración sugerida



## Material para la clase

- Acceso a *MakeCode*.
- Opcional:** Una *micro:bit* u otra tarjeta con sensor de luz y un editor de lenguaje en bloques para programarla.



## Lo que sabemos, lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

Hasta ahora has aprendido a:

- Usar arreglos para guardar información.
- Recuperar información de un arreglo.
- Realizar operaciones con esta información.
- Guardar información de imágenes en un arreglo de dos formas diferentes.

En esta sesión pondrás en práctica parte de lo aprendido.

El reto que tienes es el siguiente:

Te solicitan diseñar y probar un programa en bloques que cumpla con las siguientes **especificaciones**:

- Al oprimir el botón A el programa debe comenzar a guardar datos de nivel de luz cada 30 segundos manteniendo guardados sólo los últimos 5 valores.
- Deberá indicar cuándo ya tiene 5 valores almacenados, con alguna señal visual.
- Al oprimir el botón B deberá mostrar el promedio de los últimos 5 datos por un breve momento.

### Glosario



**Especificación:** aquello que debe cumplir un diseño.



**Restricción:** limitaciones que debe respetar un diseño.

### Recomendación

Si trabajas en simulación con *MakeCode*, deberás mover el dial de luz o de temperatura para cambiar manualmente el valor y probar el programa.

Si puedes trabajar con la tarjeta, bastará que la tapes y la destapes para cambiar la cantidad de luz que le llega.

## Manos a la obra

### Conectadas



Esta sección corresponde al 85% de avance de la sesión

Organízate en grupos de dos a tres personas siguiendo las indicaciones de tu docente y procede a responder al reto.

Antes de resolverlo, van a conocer una forma adicional de trabajar con los arreglos: como lista.

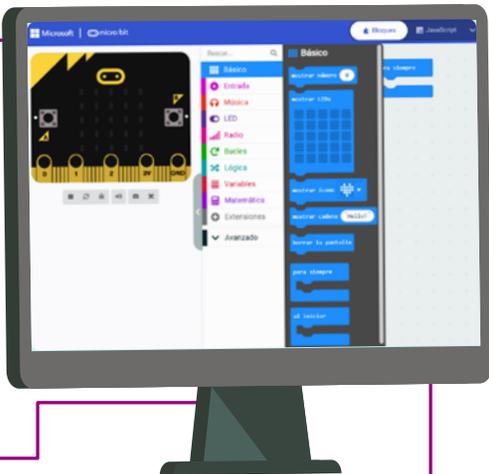
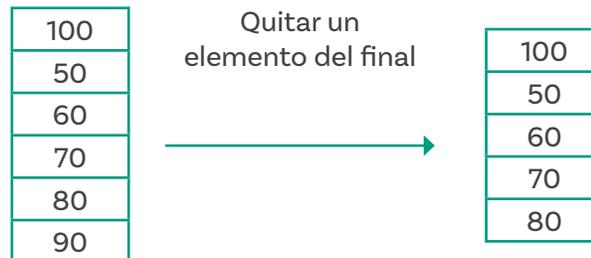
En una lista, por ejemplo, se pueden agregar elementos al comienzo y quitar elementos del final. De modo que el elemento que se quita es el más antiguo.

Deben examinar los dos ejemplos que muestran lo que pasa en una lista cuando se agrega un elemento al comienzo (Caso 1) y lo que pasa cuando se saca un elemento del final (Caso 2).

#### Caso 1



#### Caso 2



Luego estudien el siguiente programa en bloques, anticipen que hace y realicen pruebas de su funcionamiento, presentado en la *Figura 1*. Es probable que tengan que seguirlo con papel y lápiz.

**Figura 1.** Programa en MakeCode con comentarios

The image shows a MakeCode script with the following blocks:

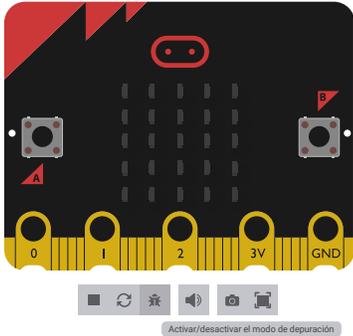
- al iniciar** (when green flag clicked):
  - Block: `fijar lista a matriz vacía` (set list to empty matrix)
  - Block: `fijar Indice a 0` (set index to 0)
- al presionarse el botón A** (when button A is pressed):
  - Block: `List insertar Indice al principio` (insert at beginning)
  - Block: `cambiar Indice por 1` (change index to 1)
  - Block: `mostrar número Indice` (show number)
- al presionarse el botón B** (when button B is pressed):
  - Block: `mostrar número obtener y eliminar el último valor de List` (show number)

Two callout boxes provide additional context:

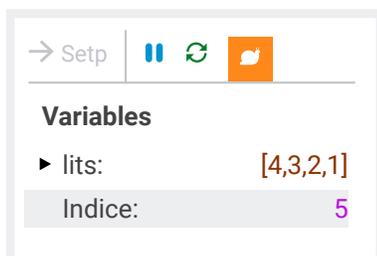
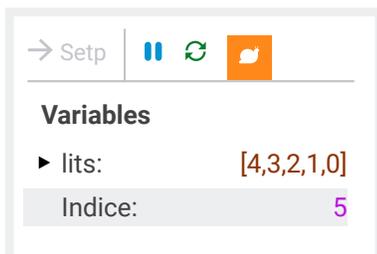
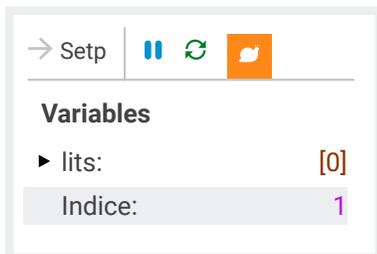
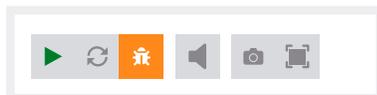
- Callout 1 (yellow):** "Se inserta un valor al comienzo del arreglo, en la posición cero, los valores existentes de desplazan hacia abajo y su longitud crece en 1" (A value is inserted at the beginning of the array, at position zero, existing values shift down and its length increases by 1).
- Callout 2 (yellow):** "Elimina el valor al final del arreglo, su longitud se reduce en 1" (Removes the value at the end of the array, its length decreases by 1).

La depuración en el lenguaje de programación MakeCode se encuentra en la parte inferior del simulador, tal y como lo puedes ver en la *Figura 2*.

**Figura 2.** Visualización de opción de simulación en *MakeCode*



**Figura 3.** Pasos para depurar en *MakeCode*



Una forma adicional de examinar que pasa en un programa es usar las herramientas de depuración.

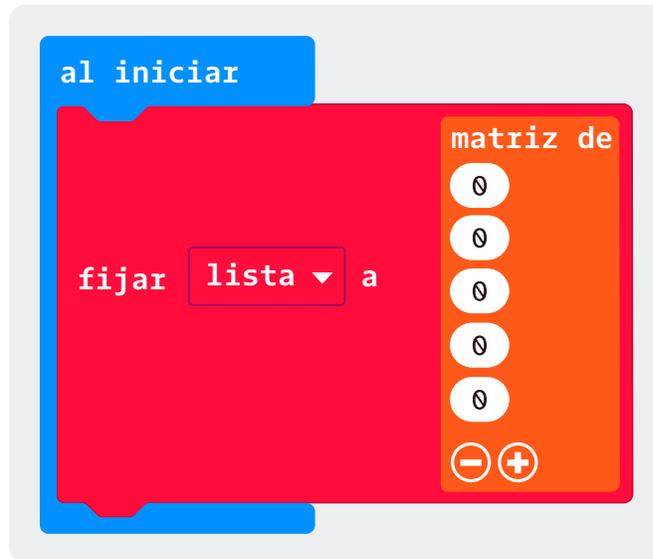
Sigan estos pasos para activar esta herramienta en *MakeCode*:

- 1 Activar el depurador en *MakeCode*. Ver *Figura 2* (el insecto).
- 2 Habilitar el caracol. Ver *Figura 3*.
- 3 Ejecutar con la flecha verde. Verán lo que muestra la *Figura 4*, después de oprimir una vez el botón A.
- 4 Aparecen las dos variables definidas en el programa, el índice que se inició en 0 y que ahora vale 1 dado que oprimiste A y el arreglo llamado lista que ya tiene un valor, el valor 0.
- 5 Si oprimen de nuevo A 3 veces más van a ver lo que aparece en la *Figura 5*. El índice habrá pasado a 5 y el arreglo llamado lista ahora tiene la secuencia 0, 1, 2, 3 y 4, siendo el valor a la derecha el que ocupa el lugar list[4].
- 6 Ahora opriman B una vez. Verán lo que aparece en la *Figura 6*. Observarán que desapareció el valor 0 que fue el primero que se incluyó y ahora el arreglo tiene solo 4 lugares.
- 7 Si continúan oprimiendo B verás desaparecer 1, luego 2, luego 3, luego 4 quedando vacío el arreglo.

Si para cada iteración colocan un valor al comienzo de un arreglo, por ejemplo, de 5 elementos y quitan el valor del final, podrán mantener guardado un número fijo de datos: los últimos 5.

Es hora de que resuelvan el pequeño reto planteado al comienzo de la sesión.

Sugerencia: comiencen inicializando un arreglo vacío de 5 elementos usando esta instrucción:

**Figura 4.** Programación de arreglo de 5 elementos

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Revisa los aprendizajes esperados de forma individual respondiendo las preguntas de forma que mejor reflejen tu progreso:

1 ¿Puedes utilizar sensores para capturar información?

- Sí
- Parcialmente
- Aún no

2 ¿Puedes organizar datos en arreglos?

- Sí
- Parcialmente
- Aún no

3 ¿Puedes utilizar la información en un arreglo para realizar cálculos?

- Sí
- Parcialmente
- Aún no

4 ¿Puedes utilizar herramientas de depuración?

- Sí
- Parcialmente
- Aún no

**Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas en *MakeCode*. 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.**

En este punto te sugerimos revisar lo que has aprendido para qué te servirá en el reto de esta guía que se encuentra en el Anexo 1.1.

Igualmente podrías tratar de identificar qué es lo que aún no sabes.

Trata de realizar un gráfico o esquema que resuma los aprendizajes logrados.

# Sesión

# 5

## Aprendizajes esperados

## Duración sugerida

Al final de esta sesión se verifica que puedas:



Utilizar sensores para capturar información.



Organizar datos en arreglos.



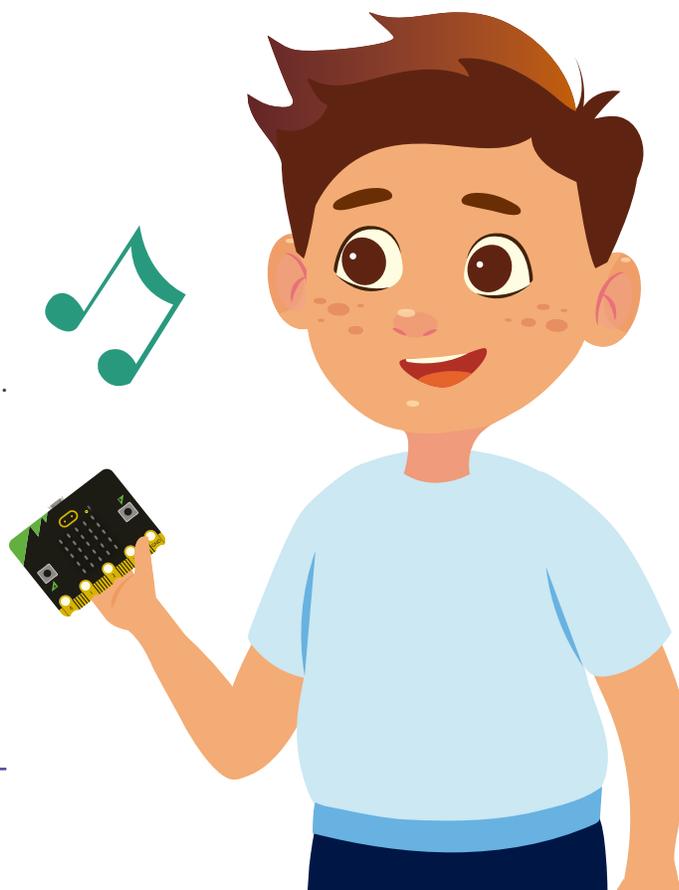
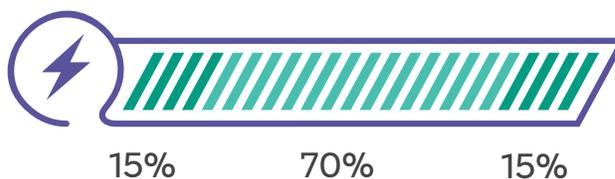
Utilizar la información en un arreglo para realizar cálculos.



Utilizar herramientas de depuración.



Reproducir notas musicales en una salida tipo parlante.



## Material para la clase

- Acceso a *MakeCode*.

**Nota**

Para el desarrollo de este reto es posible que requieras de dos sesiones de trabajo.

**Lo que sabemos,  
lo que debemos saber**



Esta sección corresponde al 15% de avance de la sesión

Comienza leyendo de nuevo el reto de esta guía, Anexo 1.1. Verás que deberás agregar señales sonoras para avisar sobre una situación de riesgo tanto en temperatura como en ruido.

Para ello usarás una de las salidas de la *micro:bit* la cuál es un parlante en el cual se pueden hacer sonar diferentes tonos.

Examina la siguiente secuencia de instrucciones y trata de anticipar el resultado.

**Anexo**

**Anexo 1.1**

**Anexo 1.1**

El monitor de variables a mostrar debe tener a la forma de decimales en función de la frecuencia con la que algo se presenta. Por ejemplo, no basta con que en un momento del día la temperatura sea un tanto elevada o un tanto sea más alta, como para que se esta situación afecte a la salud.

Usualmente la situación de la situación ambiental y el promedio de la variable medida son mejores indicadores de riesgo que una medida puntual.

En este reto te proponemos que diseñes un programa en MakeCode que permita a través de la salida de los pines de salida de la *micro:bit* avisar a través de dos grandes rangos: temperatura elevada y niveles de ruido.

Las especificaciones que debes cumplir son las siguientes:

- En el bloque de inicialización debes poderle parametrizar:
  - Se requieren variables:
    - Nivel máximo de ruido seguro y porcentaje de mediciones altas permitidas.
    - Nivel máximo de temperatura segura y porcentaje de mediciones altas permitidas.
    - Tiempo entre toma de datos.
    - Algoritmo de mediciones a realizar.
  - El programa deberá registrar la temperatura y el nivel de ruido cada intervalo definido en la inicialización.
  - Mantener almacenadas las mediciones tanto de nivel de ruido como de temperatura en el mismo archivo.
  - El algoritmo de las variables debe a partir del promedio del valor de seguridad debe emitir una señal visual y sonora que permita ser audible a la temperatura o ruido.
  - Al apretar el botón A se debe entregar el porcentaje de mediciones sobre el valor límite en temperatura.
  - Al apretar el botón B se debe entregar el porcentaje de mediciones sobre el valor límite en ruido.

El programa desarrollado deberá probarse en simulación usando MakeCode.

**Figura 1.** Secuencia de instrucciones en *MakeCode*

al presionarse el botón A

- reproduce secuencia tono **Do medio** durante 1 pulso en modo hasta que termine
- reproduce secuencia tono **Re medio** durante 1 pulso en modo hasta que termine
- reproduce secuencia tono **Mi medio** durante 1 pulso en modo hasta que termine

al presionarse el botón B

- reproduce secuencia tono **Do medio** durante 1 pulso en modo hasta que termine
- reproduce secuencia tono **Do medio** durante 1 pulso en modo hasta que termine
- reproduce secuencia tono **Do bajo** durante 1 pulso en modo hasta que termine

Luego, haz la prueba. Este pequeño programa te muestra cómo generar dos alertas sonoras diferentes. Usa tu creatividad para generar los dos avisos en caso de advertencia de riesgo por alta exposición a calor o ruido.

## Recomendación

Es posible detener por unos minutos el trabajo y que todas las personas hagan una “marcha silenciosa” examinando el trabajo de los demás grupos para inspirarse y encontrar otras formas de resolver el reto.

## Manos a la obra

### Conectadas



Esta sección corresponde al 85% de avance de la sesión

Organízate en grupo según las instrucciones de tu docente.

Es hora de resolver el reto. Como se indicó, es posible que requieras de una segunda sesión para completar con tu grupo el reto y probarlo.

Analicen en equipo el reto para identificar los elementos clave del programa en *MakeCode*. Discutan qué variables necesitarán para monitorear la temperatura y el ruido, estableciendo los límites seguros y definiendo parámetros como el intervalo entre mediciones y el número de registros que se almacenarán. Antes de programar, elaboren un plan sobre cómo se gestionarán las alertas y la visualización de los datos.

Programen la captura de temperatura y ruido en los intervalos definidos y almacenen las mediciones en una lista que conserve los datos recientes. Diseñen un sistema de alertas que active señales visuales y sonoras cuando los valores superen los límites establecidos. Implementen la funcionalidad para que, al presionar el botón A, se muestre el porcentaje de mediciones que han superado el límite en temperatura y, al presionar el botón B, se muestre el porcentaje de mediciones altas en ruido.

Prueben el programa en el simulador de *MakeCode* y verifiquen su funcionamiento.

Ajusten los parámetros si es necesario y revisen si las alertas se activan correctamente. Discutan en equipo posibles mejoras y preparen una breve demostración para compartir sus resultados con el docente y los demás grupos.

Si tu grupo tiene dificultades no duden en consultar a su docente u observar el trabajo que realizan los demás grupos.

Al final podrías mostrarle a tu docente o a otros grupos tu trabajo.

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

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

- 1 ¿Puedes utilizar los aprendizajes logrados para resolver el reto planteado?
  - Sí
  - Parcialmente
  - Aún no
- 2 ¿Puedes utilizar sensores para capturar información?
  - Sí
  - Parcialmente
  - Aún no
- 3 ¿Puedes organizar datos en arreglos?
  - Sí
  - Parcialmente
  - Aún no
- 4 ¿Puedes utilizar la información en un arreglo para realizar cálculos?
  - Sí
  - Parcialmente
  - Aún no
- 5 ¿Puedes utilizar herramientas de depuración?
  - Sí
  - Parcialmente
  - Aún no



6 ¿Puedes reproducir notas musicales en una salida tipo parlante?

- Sí
- Parcialmente
- Aún no

**Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas en *MakeCode*. 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.**

Con tu grupo reflexiona sobre las siguientes preguntas:



*¿En dónde estuvo la mayor dificultad? ¿Por qué creen que fue difícil?*

*¿Se te ocurren otros problemas que se puedan resolver utilizando lo que aprendiste?*

Podrás hacer un esquema en que quede el ejemplo que realizaste.



### Computación y sociedad

La computación se utiliza cada vez con más frecuencia para ayudarle a las personas a detectar situaciones de riesgo, en este caso, temperatura o ruido altos.

Para desarrollar todas estas herramientas se requieren personas que sepan de computación, por lo que este tipo de profesional es cada vez más solicitado.

## Anexo 1.1 Reto



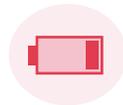
Dolor de cabeza aunque sea leve



Decaimiento y desánimo



Somnolencia



Fatigabilidad



Apatía

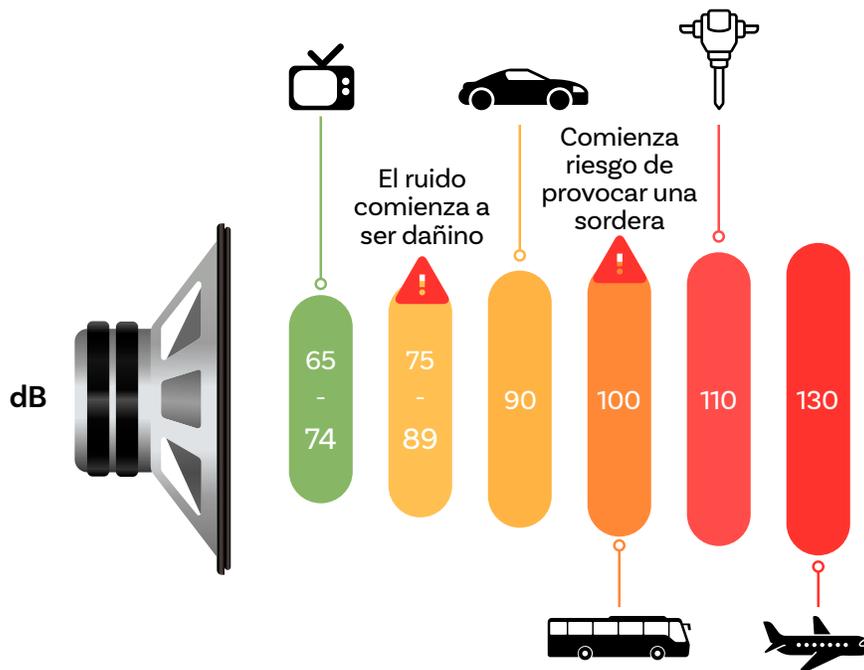


Pocas ganas de alimentarse e hidratarse

El monitoreo de variables a menudo debe llevar a la toma de decisiones en función de la frecuencia con la que algo se presente. Por ejemplo, no basta con que en un momento del día la temperatura sea un tanto elevada o un ruido sea muy alto, como para que se esta situación afecte a la salud.

Usualmente la duración de la situación anómala y el promedio de la variable medida son mejores indicadores de riesgo que una medida puntual.

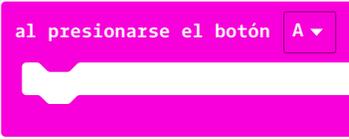
En este reto te proponemos que diseñes un programa en *MakeCode* orientado a cuidar la salud de las personas mayores frente a dos grandes riesgos: temperaturas elevadas y exceso de ruido.



Las especificaciones que debe cumplir son las siguientes:

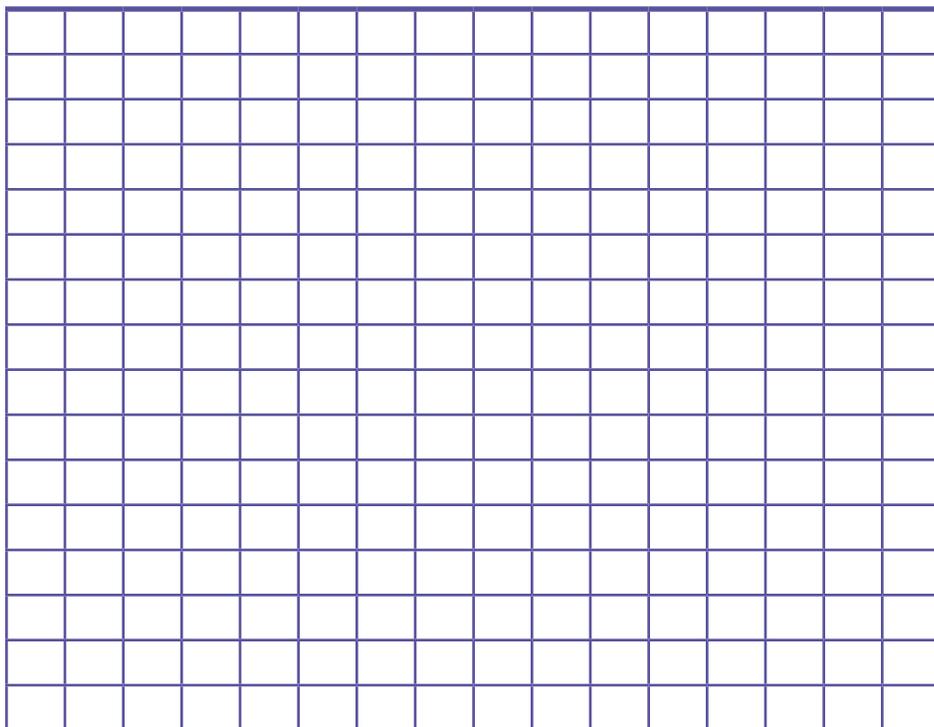
- En el bloque de inicialización deben poderse parametrizar las siguientes variables:
    - Nivel máximo de ruido seguro y porcentaje de mediciones altas permitidas.
    - Nivel máximo de temperatura segura y porcentaje de mediciones altas permitidas.
    - Tiempo entre toma de datos.
    - Número de mediciones a mantener.
  - El programa deberá registrar la temperatura y el nivel de ruido cada intervalo definido en la inicialización.
  - Mantener almacenadas las mediciones tanto de nivel de ruido como de temperatura en el número definido.
  - Si alguna de las variables llega a estar por encima del valor de seguridad debe emitirse una señal visual y sonora claramente identificable si es temperatura o ruido.
  - Al oprimir el botón A se debe entregar el porcentaje de mediciones sobre el valor límite en temperatura.
  - Al oprimir el botón B se debe entregar el porcentaje de mediciones sobre el valor límite en ruido.
- El programa desarrollado deberá probarse en simulación usando *MakeCode*.

## Anexo 1.2 ¿Salida o entrada?

Instrucción	Explicación	¿Es una entrada?	¿Es una salida?
	Muestra un corazón en la pantalla de LED de la <i>micro:bit</i> .		
	Realiza una acción si el botón A se oprime.		
	Realiza una acción si la <i>micro:bit</i> se mueve.		
	Hace sonar un tono en do medio.		
	Reproduce una melodía a partir de datos.		
	Enciende un LED particular.		
	Borra la pantalla.		
	Dato de dirección de la brújula en la <i>micro:bit</i> .		
	Nivel de luz medido por la <i>micro:bit</i> .		

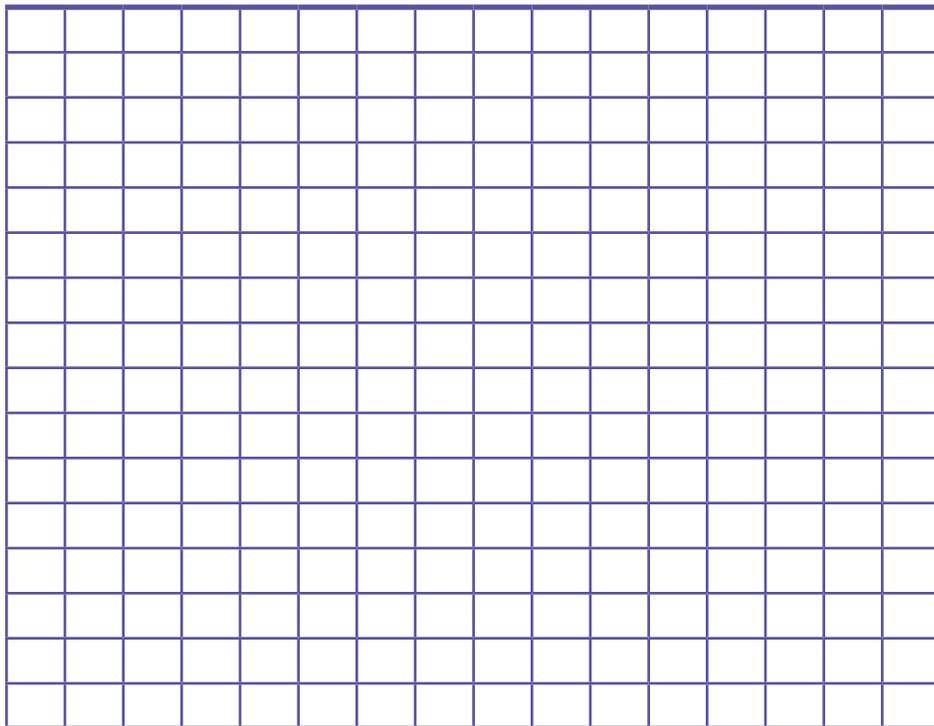
**Anexo 3.1** Imágenes codificadas

Primera imagen



- 5, 4, 7
- 4, 6, 6
- 3, 8, 5
- 2, 10, 4
- 3, 1, 1, 12, 1, 1, 1, 5
- 3, 1, 1, 12, 1, 1, 1, 5
- 3, 1, 1, 12, 1, 1, 1, 5
- 3, 1, 1, 12, 1, 1, 1, 5
- 3, 8, 5
- 3, 12, 1
- 3, 12, 1
- 3, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1
- 3, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1
- 0, 6, 2, 3, 3, 1, 1
- 0, 6, 2, 3, 3, 1, 1
- 0, 16

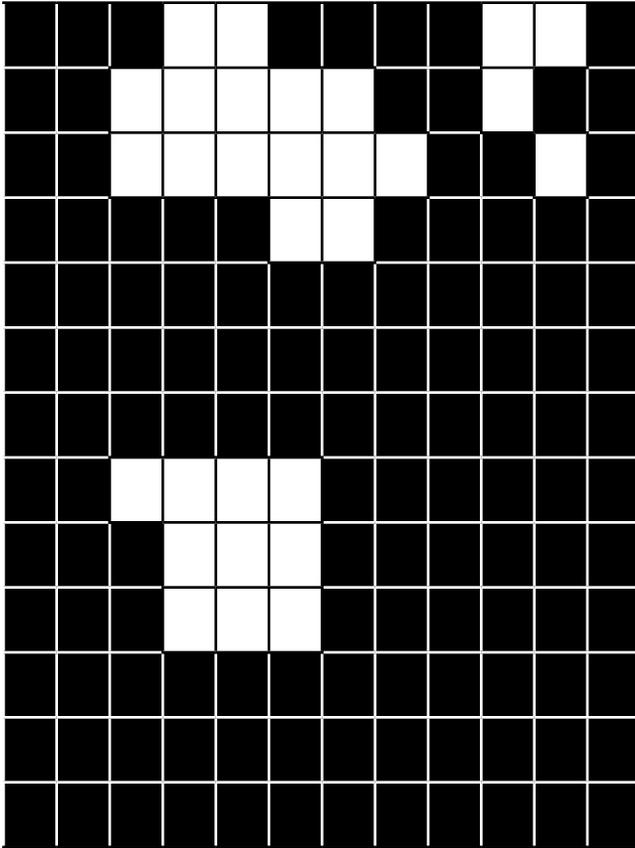
Segunda imagen



- 1, 1, 14
- 0, 2, 14
- 0, 2, 14
- 1, 1, 14
- 1, 1, 14
- 1, 1, 2, 2, 10
- 1, 1, 1, 4, 9
- 1, 7, 8
- 1, 8, 7
- 1, 9, 6
- 1, 10, 5
- 1, 11, 4
- 2, 8, 1, 2, 3
- 2, 1, 1, 3, 1, 1, 3, 2, 2
- 2, 1, 5, 1, 4, 2, 1
- 2, 1, 5, 1, 5, 2

Anexo 3.2 Imagen codificada del glaciar del Cocuy en dos momentos

Glaciar del Cocuy ahora



Glaciar del Cocuy ahora

- 0, 2, 6, 1, 3
- 0, 2, 10
- 0, 3, 9
- 0, 4, 5, 1, 1, 1
- 0, 4, 4, 4
- 0, 1, 3, 2, 4, 2
- 0, 5, 5, 2
- 0, 1, 7, 4
- 0, 3, 5, 4
- 0, 3, 5, 4
- 0, 2, 6, 4
- 0, 4, 4, 4
- 0, 5, 3, 4

Utilizando una hoja cuadriculada encontrar la imagen del glaciar del Cocuy hace 20 años y determinar el porcentaje perdido de glaciar en 20 años.





# TIC



Apoya:



**Educación**



**BRITISH  
COUNCIL**



**Colombia  
Programa**

{EL CÓDIGO A TU FUTURO}