Salvando a las tortugas: bucles

Grado 5° Guía 2

TIC

Estudiantes

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

Salvando a las tortugas: bucles

Grado 5° Guía 2

TIC

Estudiantes

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

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 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 5° Guía 2

TIC

Guía de íconos

- Algoritmos, patrones, abstracción y descomposición
- 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:

- Subdividir tareas complejas en otras más sencillas.
- Identificar grupos de pasos que se repiten en un algoritmo o programa y reconocerlos como oportunidades para el uso de bucles.
- Usar diagramas para representar algoritmos que incluyen bucles y condicionales.
- Usar variables físicas en algoritmos.

Resumen de la guía

Esta guía te propone 5 sesiones de trabajo que te ayudarán a recordar el concepto de bucles o ciclos para controlar la ejecución de algoritmos y programas sencillos. Las actividades planteadas te permitirán seguir mejorando tus habilidades de reconocimiento de patrones y te permitirán identificar oportunidades de uso de los bucles para resumir conjuntos de instrucciones.

Resumen de las sesiones

Sesión 1

En esta sesión se afianza el concepto de bucle a través de la simplificación de instrucciones orales y escritas en algoritmos varios.

Sesión 2

Se presentan los diagramas de flujo como una forma de escribir y representar instrucciones. A través de actividades desconectadas, la clase practicarás la lectura de diagramas de flujo que incluyen bucles.

Sesión 3

Se transfiere el concepto de bucles al editor *MakeCode*. Se aprende a utilizar bloques para repetir instrucciones en los programas.

Aprendizajes de la guía

- Usar el sensor de temperatura para recuperar o transmitir datos a un microprocesador.
- Crear y depurar programas que incluyan bucles de diferentes tipos (repetir, mientras, para siempre).

Sesión 4

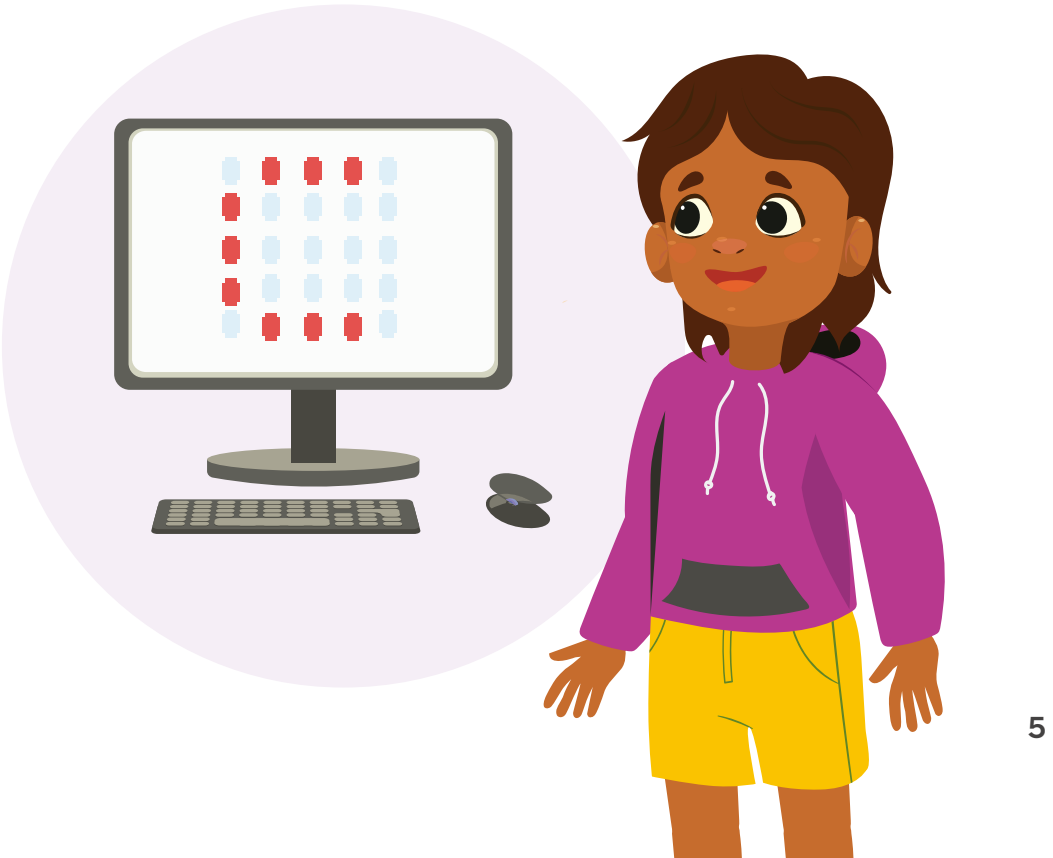
Se continúa usando bucles en *MakeCode*, pero ahora se controlan los bucles con otras variables de entrada como los botones y el sensor de temperatura.

Sesión 5

Se aplica lo aprendido para la solución de una problemática realista.

Nota

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



Grado 5° Guía 2

TIC

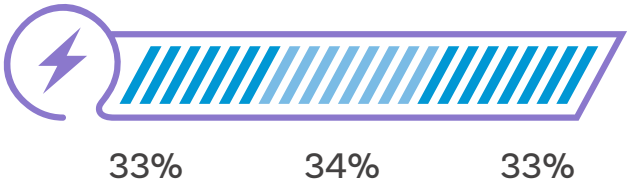
Sesión 1

Aprendizajes esperados

Al final de esta sesión se espera que puedas:

- Descomponer un algoritmo o programa en segmentos o grupos de pasos que se repiten.
- Explicar los diferentes tipos de bucles que se utilizan en programación.
- Simplificar un programa reduciendo pasos o instrucciones repetidas mediante el uso de bucles.

Duración sugerida

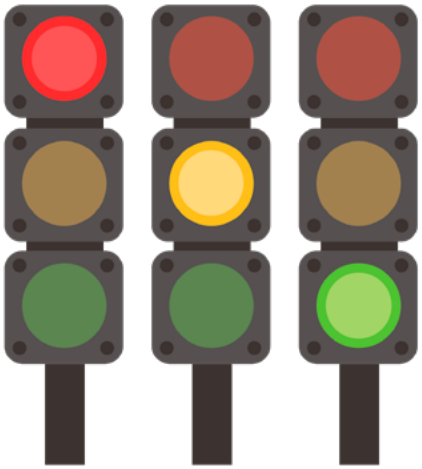


Material para la clase

- Anexos 1.1, 1.2, 1.3
- 12 fichas

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 33% de avance de la sesión



A diario te puedes encontrar con situaciones y circunstancias en las que puedes identificar acciones y eventos que se repiten en más de una ocasión. Estos casos permiten ejemplificar lo que en programación se conoce como **bucles**. El recorrido del sol a lo largo del día, desde el amanecer hasta el atardecer, por ejemplo, se repite día tras día, en el mismo orden, por lo que puede considerarse un ejemplo de **bucle infinito** en la naturaleza. Ponerse los zapatos, por ejemplo, es una acción que se repite solo 2 veces, pues te pones un zapato, luego el otro y con eso se completa la tarea. Este es un ejemplo de **bucle repetir un número de veces**. Finalmente, hay acciones que dependen de que se cumpla una condición. Por ejemplo, piensa en el semáforo. Como sabes, este dispositivo de control de tráfico cambia los colores de las luces en un orden específico y cada uno de estos colores indica un comportamiento esperado: mientras la luz verde esté encendida, los vehículos pueden avanzar. En cambio, mientras la luz roja esté encendida, deben quedarse en su lugar y esperar. Este es un buen ejemplo de **bucle condicional** o **bucle mientras que**.

Ahora vas a realizar un ejercicio que te ayudará a comprender mejor las acciones que se realizan solo si se cumple alguna condición.

Trabaja en grupo con 2 o 3 personas más, según lo indique tu docente. Junto a tus compañeras y compañeros, definan tres movimientos sencillos que puedan hacer en sus puestos y asócienlos a cada color del semáforo. Por ejemplo, tú puedes decir: “Mientras yo diga que el semáforo está en rojo, ustedes van a agitar la cabeza como diciendo No”, o “Mientras yo diga que el semáforo está en amarillo, ustedes sacuden las dos manos”.

Cuando tengan los movimientos decididos, una persona hará las veces de semáforo diciendo colores al azar. Por ejemplo:

"Amarillo... amarillo... amarillo... verde" y los demás deben ejecutar las acciones que correspondan y repetirlas hasta que la persona que dirige elija otro color. Jueguen durante dos minutos y cambien los roles. Al finalizar, discutan:

¿?

¿Fue difícil seguir las secuencias?
¿Cuál fue la instrucción más difícil de realizar?
¿Qué pasó cuando el semáforo cambió de rojo a verde, sin pasar por amarillo?
¿Es posible saber cuántas veces hicieron una acción específica?

Así como en el ejercicio anterior, a veces se tienen que repetir acciones mientras se cumple una condición, o hasta que se complete una secuencia.

Piensa ahora en otra situación. Considera, por ejemplo, las acciones requeridas para doblar una camiseta, las cuales pueden verse así:

1. Tomo la camiseta.
2. Doblo la manga izquierda hacia el centro.
3. Doblo la manga derecha hacia el centro.
4. Doblo la parte de arriba hacia abajo.
5. Suelto la camiseta.

Pero si tienes muchas camisetas por doblar, el proceso podría verse así:

Mientras tenga camisetas por doblar...

1. Tomo la camiseta.
2. Doblo la manga izquierda hacia el centro.
3. Doblo la manga derecha hacia el centro.
4. Doblo la parte de arriba hacia abajo.

5. Suelto la camiseta.
6. Tomo la camiseta.
7. Doblo la manga izquierda hacia el centro.
8. Doblo la manga derecha hacia el centro.
9. Doblo la parte de arriba hacia abajo.
10. Suelto la camiseta.
11. Tomo la camiseta.
12. Doblo la manga izquierda hacia el centro.
13. Doblo la manga derecha hacia el centro.
14. ...

¿?

¿Puedes identificar cómo se repiten las instrucciones?

Una forma más sencilla de escribir lo mismo sería:

Mientras haya camisetas por doblar:

1. Tomo la camiseta.
2. Doblo la manga izquierda hacia el centro.
3. Doblo la manga derecha hacia el centro.
4. Doblo la parte de arriba hacia abajo.
5. Suelto la camiseta.
6. Me pregunto si hay más camisetas por doblar.

¿Notas cómo el uso de **bucles mientras que** permite simplificar las instrucciones?

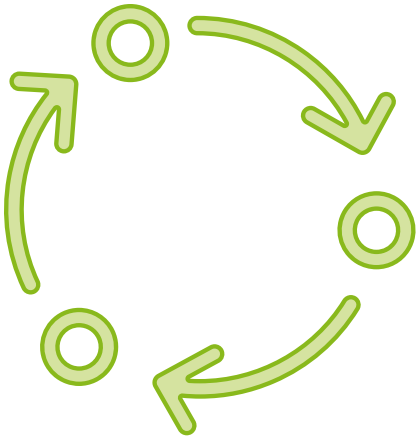
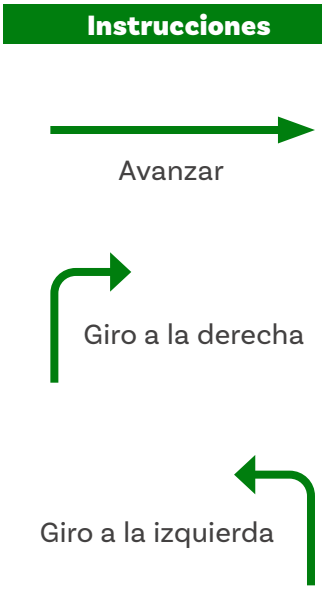


Figura 1. Instrucciones para representar los movimientos de las tortugas



Glosario

- **Bucles:** en programación, se llama bucles o ciclos a los bloques de instrucciones que se repiten una y otra vez, mientras se cumpla una condición.
- **Bucles infinitos:** son aquellos en los que las instrucciones dentro del bucle se repiten para siempre.
- **Bucles mientras que:** las instrucciones se repiten mientras algo sucede. Por ejemplo: “Mientras esté en la escuela, estudio, interactúo con mis compañeras y compañeros”.
- **Bucles repetir un número de veces:** las instrucciones se repiten un número exacto de veces. Por ejemplo: “Mi canción favorita repite tres veces el coro”.

Manos a la obra

Desconectadas

Esta sección corresponde al 67% de avance de la sesión

Vas ahora a realizar una actividad que te permitirá aplicar lo aprendido sobre bucles. Tal como en el nombre de esta guía, tu trabajo se relacionará con las tortugas. Estos hermosos animales marinos visitan las costas del Pacífico y del Caribe colombianos para poner sus huevos.

Tu reto será escribir el algoritmo que deben seguir dos tortugas desde el lugar donde se encuentran hasta el nido donde depositarán sus huevos. Piensa que cada flecha es un paso que pueden dar las tortugas. Usa tu cuaderno para escribir cada instrucción del algoritmo siguiendo los símbolos que aparecen en la *Figura 1*. Luego, revisa cómo puedes simplificar cada algoritmo usando bucles. Fíjate en el ejemplo que se presenta en la *Figura 2*.

Figura 2. Ejemplo de bucle

Paso a paso	Usando bucles
Ir adelante Ir adelante Ir adelante Ir adelante	Repetir 4 veces: Ir adelante

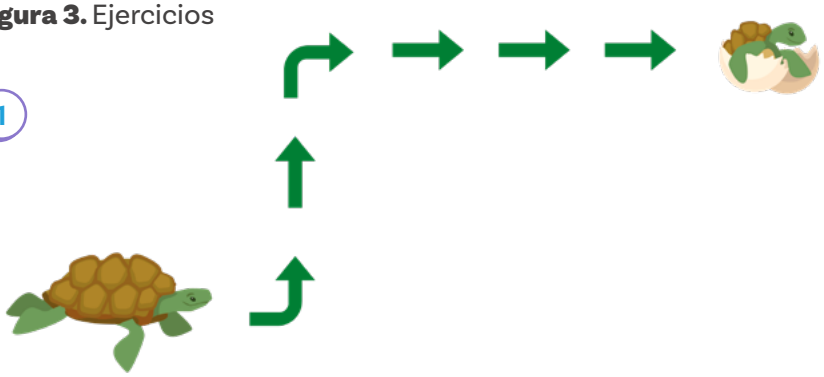
¿Qué tipo de bucle es el que se usó en el ejemplo anterior?

¿Se habría podido usar otro tipo de bucle en vez de ese? ¿Por qué?

Ahora sí, tus ejercicios:

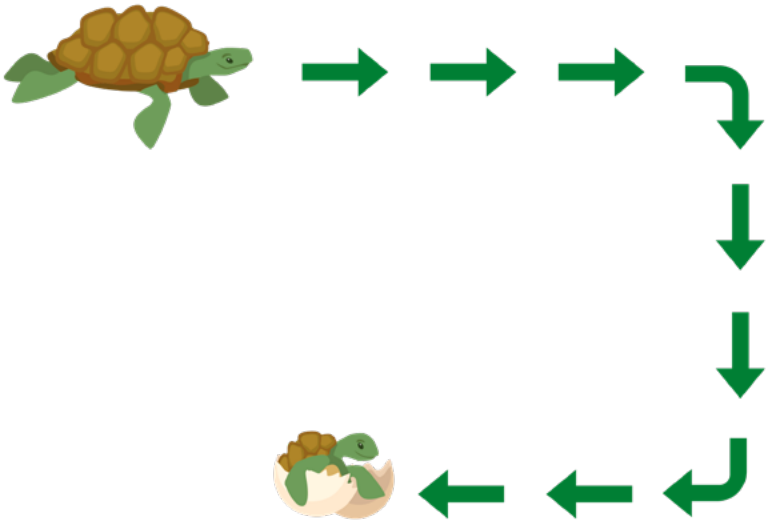
Figura 3. Ejercicios

1



Paso a paso	Usando bucles

2



Paso a paso	Usando bucles

Para ir más lejos

Sigue trabajando con el grupo de estudiantes con el que hiciste la discusión anterior. Imaginen que esta vez, para llegar al nido, la tortuga debe realizar un recorrido que se asemeja al que seguirían ustedes al caminar alrededor de su salón de clase.

¿?

¡¿Cómo se verían las instrucciones para recorrer ese espacio?

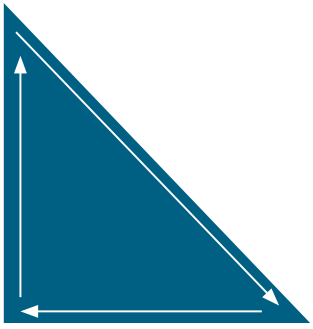
¿Qué pasos se repetirían?

El ejemplo de la *Figura 4* puede servirles de inspiración para la creación de su algoritmo. No duden en pedir apoyo a su docente si tienen alguna dificultad para plantear su propuesta.

En el algoritmo de la Figura se utilizan pasos como indicador de avance, pero ustedes pueden pensar otras opciones: metros, baldosas o incluso guiarse por elementos en su salón. Usen su creatividad para llegar al objetivo.

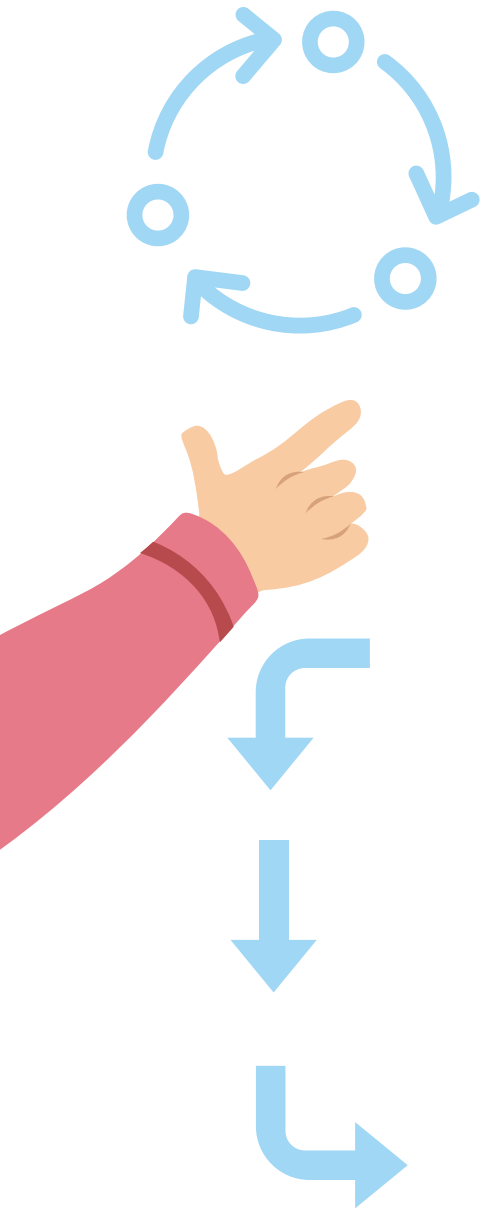
Figura 4. Algoritmo para recorrer el salón

Si tu salón tuviera la forma de un triángulo, los pasos podrían verse así si comienzas de abajo hacia arriba.



- Repetir 10 veces:
  Dar un paso hacia adelante
- Girar 120° hacia la derecha

- Repetir 10 veces:
  Dar un paso hacia adelante
- Girar 120° hacia la derecha
- Repetir 10 veces:
  Dar un paso hacia adelante



Comparte a dos de tus compañeras o compañeros las instrucciones del algoritmo que construiste.

¿?

¿Identificaron qué pasos se repiten?
¿Dividieron el camino de la tortuga en segmentos más pequeños para encontrar posibles soluciones?

¿Se les ocurren otras formas de escribir las instrucciones del algoritmo?

¿Qué tipo de bucle es el que se usa en cada caso?

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

Los ejercicios anteriores, de llevar las tortugas hasta su lugar de desove, requerían soluciones con bucles o ciclos, del tipo: bucle que repite un conjunto de instrucciones por un número de veces.

Piensa en esta pregunta:

¿Puedes reemplazar este tipo de bucle que se repite un número de veces por uno que se realice mientras se cumpla alguna condición?

¿Has jugado a la silla musical? En este juego, mientras suena la música, se camina o baila alrededor de un número de sillas que es menor al número de personas jugando, y cuando la música se detiene todas las personas se sientan. Quien no logre sentarse, sale del juego. ¿Cómo escribirías las reglas de este juego utilizando bucles?

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

1. ¿Puedes descomponer un algoritmo o programa en segmentos o grupos de pasos que se repiten?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes explicar los diferentes tipos de bucles que se utilizan en programación?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

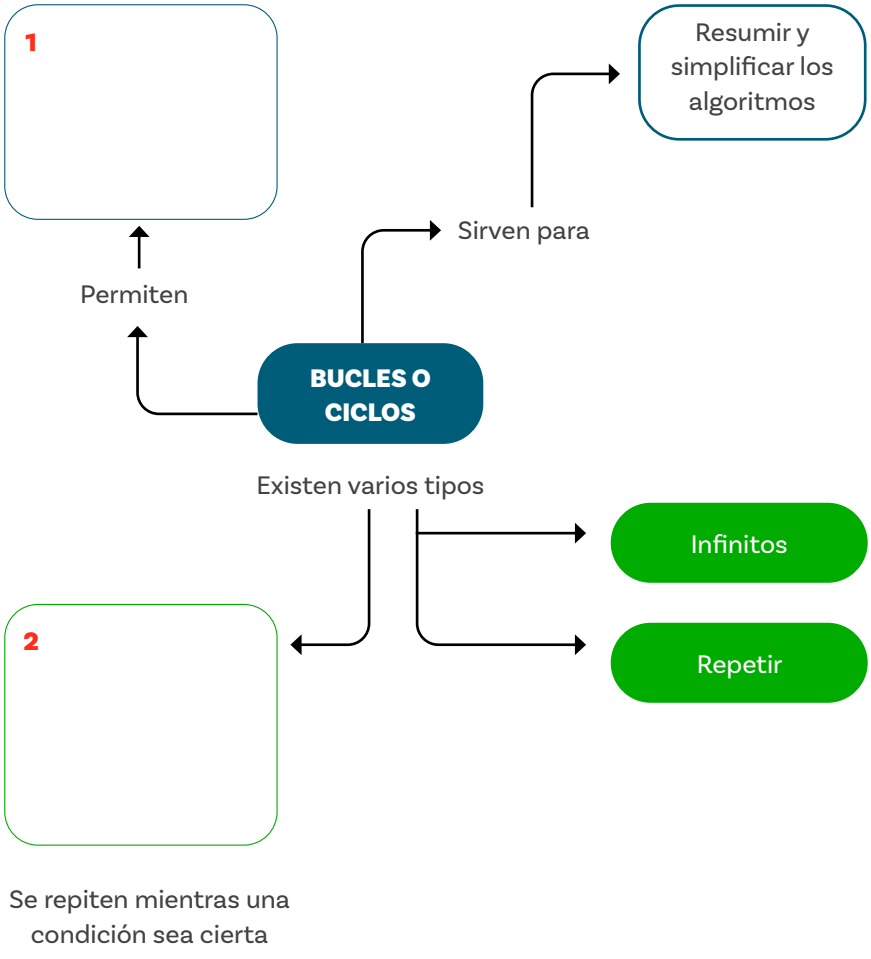
3. ¿Puedes simplificar un programa reduciendo pasos o instrucciones repetidas mediante el uso de bucles?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", lee nuevamente el glosario, revisa los ejemplos de bucles que se presentan al inicio de la sesión, conversa con tus compañeras y compañeros de grupo en las actividades anteriores, y si después de todo esto todavía tienes dudas, consulta con tu docente. Una vez hayas aclarado las dudas, dibuja otro ejemplo de trayectoria de una tortuga hasta su nido, divide los pasos del camino en elementos comunes o repetidos y escribe el algoritmo correspondiente usando bucles. Compártelo con tu docente.

Ahora, completa las palabras faltantes en el gráfico de resumen de la *Figura 5*.

Figura 5. Gráfico de resumen



Verifica las respuestas con ayuda de tu docente.

Sesión 2

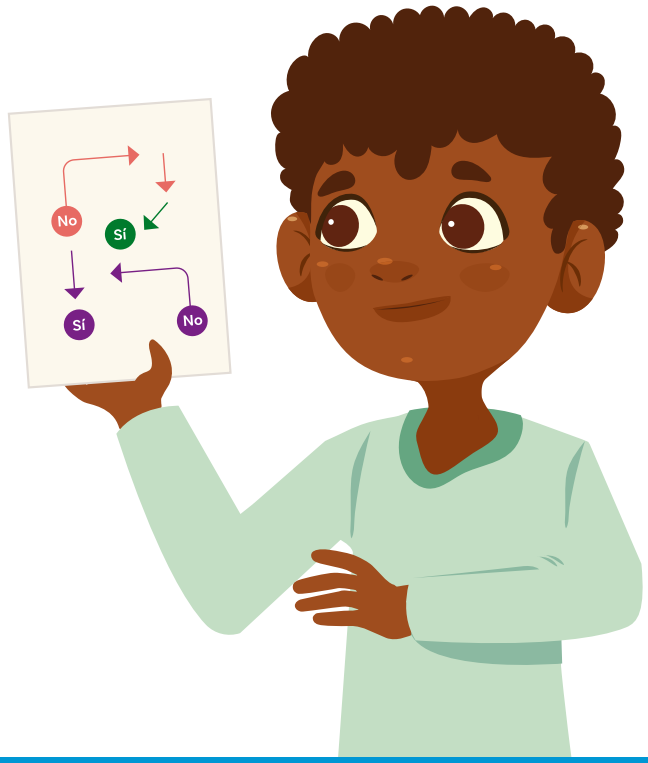
Aprendizajes esperados

Al final de esta sesión se espera que puedas:

- Leer y ejecutar algoritmos representados mediante diagramas de flujo.
- Representar, mediante diagramas de flujo, algoritmos que incluyan bucles finitos y condicionales.
- Depurar algoritmos que incluyan bucles finitos y bucles condicionales.

Duración sugerida

15% 70% 15%



Material para la clase

- ○ Anexos 2.1 y 2.2

Nota: existen diferentes lenguajes que sirven para programar distintos procesadores, pero todos describen un proceso lógico de pasos o instrucciones. Algunas acciones requieren pocos pasos, como, por ejemplo, mostrar un corazón en el panel LED de la micro:bit, mientras que otros pueden requerir miles y miles de pasos, Por eso, es útil utilizar bucles o ciclos que permitan simplificar instrucciones que se repiten.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En la sesión anterior encontraste que, en un trayecto o grupo de movimientos como los que hacían las tortugas de la actividad, había elementos comunes que se repetían. En esta sesión vas a seguir ampliando lo aprendido en cuanto al uso de algoritmos que incorporan bucles, pero esta vez lo harás a partir de otras situaciones.

Mira el siguiente patrón. ¿Qué parte se repite? ¿Cuántas veces se repite? ¿Podrías colorear el siguiente banderín?

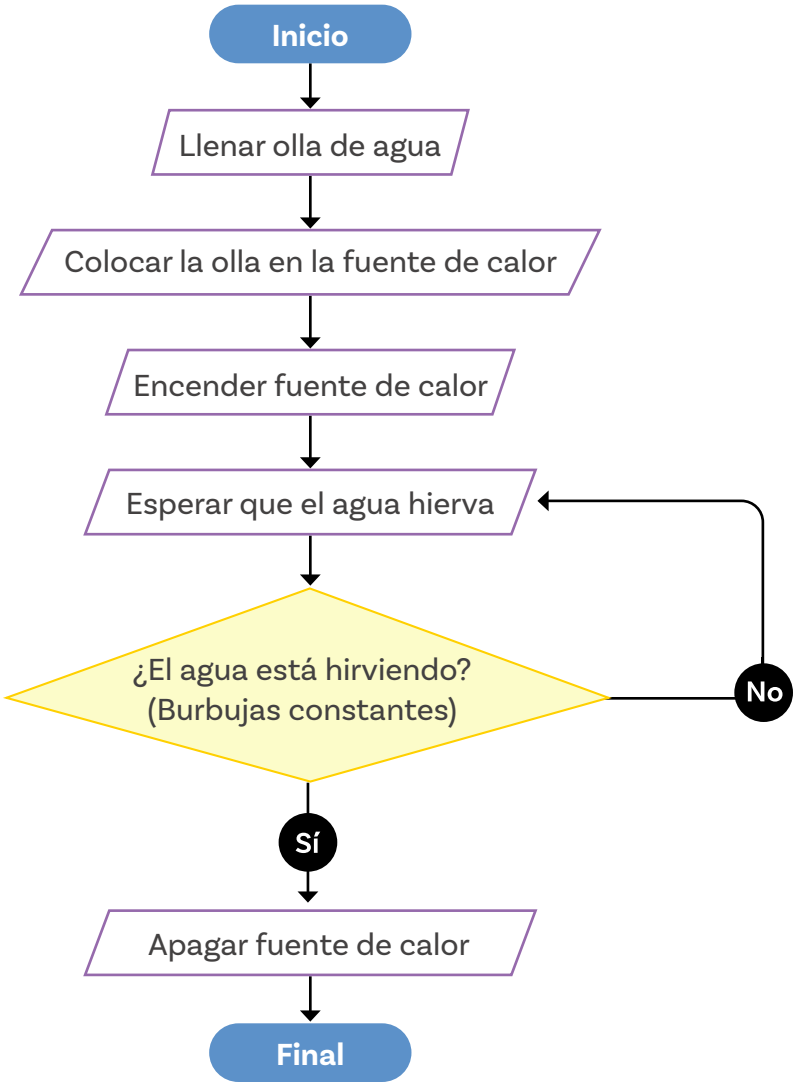
Figura 1. Banderines de colores

En la *Figura 1* vemos siete banderines, ¿puedes adivinar de qué color será el banderín número 15?

Para predecir el banderín que sigue, utilizas una habilidad llamada **reconocimiento de patrones**. En esta sesión, la habilidad de reconocer patrones va a ser muy importante para identificar en dónde puedes utilizar bucles.

Pero antes, vas a aprender la forma de escribir y representar instrucciones mediante **diagramas de flujo**. El ejemplo de la *Figura 2* te mostrará cómo un proceso cualquiera como, por ejemplo, hervir agua, puede expresarse de esta manera. Piensa en los pasos que seguirías para hervir agua.

Figura 2. Diagrama de flujo



¿Qué pasos sigues cuando vas a hervir agua?

Ten presente que se inicia con alguna acción: llenar la olla de agua, por ejemplo y, usualmente, hay un momento en que se toma una decisión en función de la temperatura del agua. En el ejemplo de la *Figura 2*, si el agua no ha llegado a la temperatura que se desea, se sigue el proceso. Pero si ya lo hizo, se finaliza. La flecha que se devuelve indica que el proceso se repite hasta que se cumpla la condición de la temperatura deseada, así que esa flecha representa un bucle.

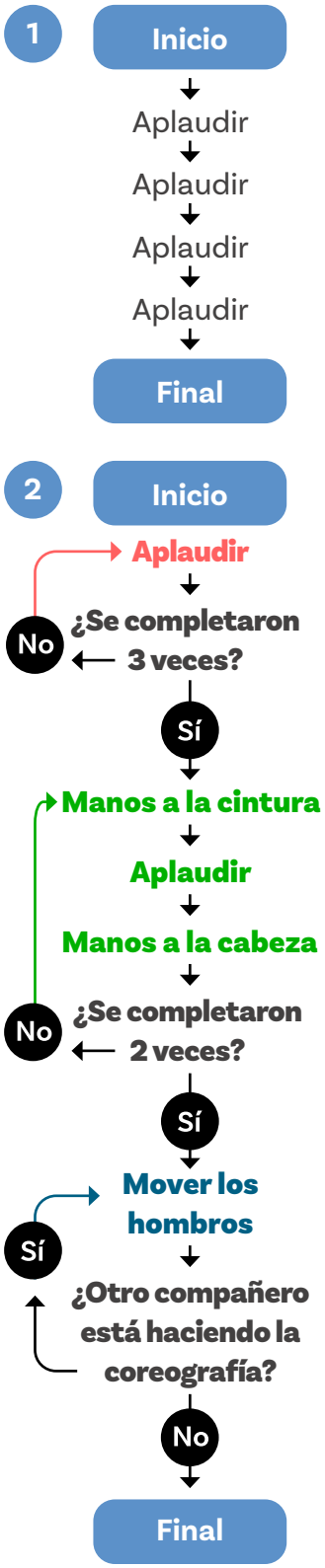
Piensa ahora en otro proceso sencillo que hagas en tu vida cotidiana. Puede ser algo como tender la cama, atarte los cordones, limpiar el piso o incluso comerte una manzana.

¿Puedes expresar este proceso mediante un **diagrama de flujo?**

Glosario

- **Diagrama de flujo:** gráfico que representa un algoritmo e indica el paso a paso para resolver una tarea o completar un proceso.
- **Reconocimiento de patrones:** habilidad para encontrar elementos comunes, repetitivos o con características semejantes dentro de problemas, códigos, conjuntos de datos, etc.
- **Depurar:** en programación se le llama depurar al proceso de identificar y corregir errores en los algoritmos o programas.

Figura 3. Ejemplos diagramas de flujo



Manos a la obra

Desconectadas

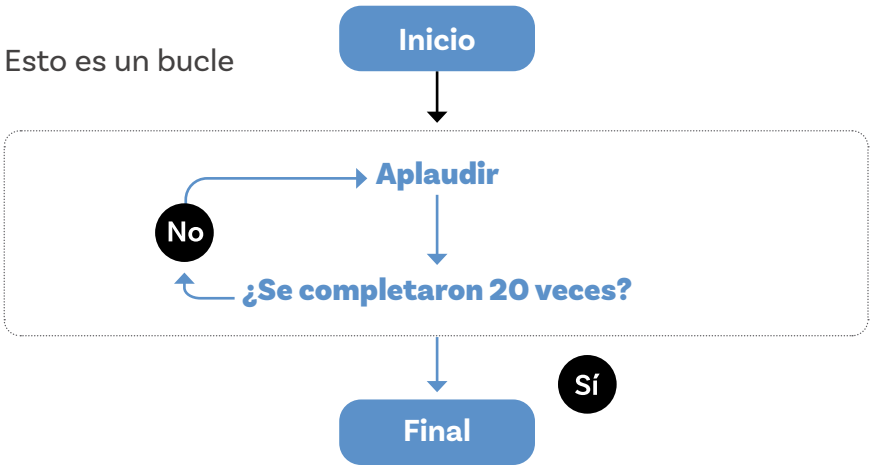
Esta sección corresponde al 85% de avance de la sesión

La siguiente actividad te permitirá seguir ahondando la comprensión sobre el uso de diagramas de flujo. Observa los dos ejemplos que se presentan en la *Figura 3*. Luego, trabaja estos diagramas de flujo con alguien más de tu salón, siguiendo las indicaciones de tu docente.

1. Ejecuten el algoritmo del diagrama de flujo identificado con el número 1.

2. Si tuvieran que aplaudir muchas veces, por ejemplo 20, podrían colocar 20 veces la instrucción Aplaudir, o simplemente representarlo mediante un bucle como el que se muestra a continuación.

Esto es un bucle

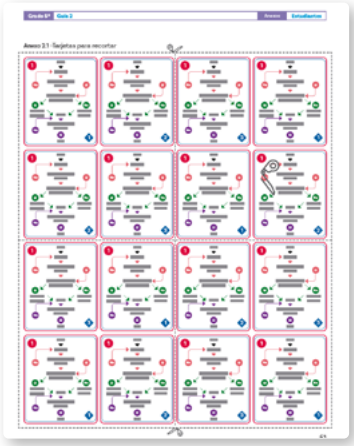


3. Ahora sigan el algoritmo del diagrama de flujo identificado con el número 2, que representa una coreografía que deben ejecutar varias personas hasta terminar. ¿Cómo les fue siguiendo este algoritmo? ¿Les funcionó?

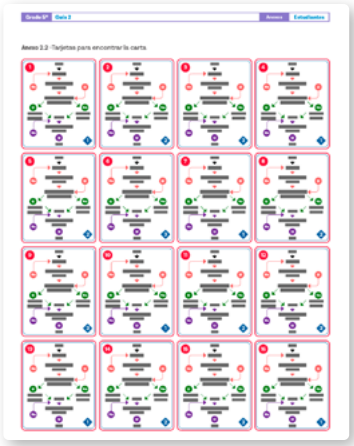
4. Si algo les parece incorrecto, tienen razón. Se requiere que una persona depure el código para corregirlo. ¿Qué se podría hacer? Discútanlo como equipo y, si es preciso, discútanlo con ayuda de su docente.

Anexos

Anexo 2.1



Anexo 2.2



5. Habrán encontrado que algunos bucles se repiten 3 veces o 2 veces, pero también aparece un bucle diferente que se repite mientras esta condición sea cierta: ¿otro compañero o compañera está haciendo la coreografía?

6. Este bucle se repetirá mientras la condición sea cierta y, como en la clase tienen muchas compañeras y compañeros, probablemente tengan que quedarse moviendo los hombros indefinidamente y el resto del grupo también seguirá haciéndolo dado que ustedes también lo hacen.

7. ¿Ya encontraron una solución a este problema? Si la encuentran, están haciendo la depuración del algoritmo.

Es el momento de aplicar lo que aprendieron. Van a jugar con unas tarjetas poco usuales, que se encuentran en el *Anexo 2.1*. Cada una describe una coreografía en forma de diagrama de flujo. Su misión será demostrar la coreografía que les asignen y tratar de descubrir la coreografía que realicen otros grupos con el fin de ganar muchos puntos. Las instrucciones del juego son las siguientes:

1. Organícense en grupos de 3 a 5 integrantes, según les indique su docente.

2. Recibirán una copia completa del *Anexo 2.2* sin recortar.

3. Su docente les invitará a pasar a tomar una tarjeta recortada del *Anexo 2.1*. Háganlo de modo que los demás grupos no sepan cuál tarjeta eligieron.

4. Como equipo lean las instrucciones de la tarjeta que tomaron y asegúrense de que entienden cómo será la coreografía que deberán demostrar, según el diagrama de flujo en su tarjeta.

5. Escojan a una persona que tenga buena memoria para que pase al frente a realizar la coreografía descrita en la tarjeta que seleccionaron.

6 Mientras su compañera o compañero del grupo pasa y hace la coreografía, los otros grupos tendrán 1 minuto para buscar entre todos los diagramas de flujo cuál representa la coreografía de su grupo y escribir el número en un papel. Tengan en cuenta que este número se encuentra en la esquina superior izquierda de la tarjeta, dentro de un círculo.

El segundo condicional en el algoritmo se refiere al grupo que presenta la coreografía. A la señal de su docente, muestren todas las respuestas. Si alguien habla en voz alta dando la respuesta antes de esta señal, pierde 2 puntos.

Su docente asignará los puntos indicados en la esquina inferior derecha de cada tarjeta a quienes hayan encontrado la solución correcta. El grupo que hizo la representación también recibe ese puntaje. Si ningún grupo encuentra el diagrama correcto nadie recibe puntos.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

Dibuja el diagrama de flujo de alguna actividad que realices a diario y prepárate para compartirlo con otras personas de tu clase. No olvides incluir en tu diagrama las acciones que se deban repetir mediante bucles finitos o condicionales.

Algoritmo para: _____

Piensa un momento en las siguientes preguntas, discútelas con una compañera o compañero y luego respondan:

¿De qué forma se representaban los bucles en los diagramas de flujo que observaron?

¿Para qué puede servir hacer un diagrama de flujo?

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

1. ¿Puedes leer y ejecutar algoritmos representados mediante diagramas de flujo?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes representar, mediante diagramas de flujo, algoritmos que incluyan bucles finitos y condicionales?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

3. ¿Puedes depurar algoritmos que incluyan bucles y condicionales?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ No

Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", recuerda que los bucles finitos son los que se repiten un número de veces. Es decir que cada vez que encuentres y sigas la indicación de repetir dos o más veces las acciones de un algoritmo, estás ejecutando o haciendo uso de bucles finitos. Además, ten presente que, si realizas una acción sólo mientras se cumple una condición o hasta que esta se dé, estás haciendo uso de bucles condicionales. Ya que tienes esto más claro, revisa nuevamente las actividades realizadas en la clase y discute con tu docente cualquier otra inquietud que aún tengas sobre los temas de esta sesión. Luego, representa mediante un diagrama de flujo, al menos, un fragmento de tu propia versión de la coreografía de la canción "La Tortuga" de Joe Arroyo, o la canción "Tortuguita, vení y bailá" que es una canción de la tradición oral del pacífico colombiano.

Grado 5° Guía 2

TIC

Sesión 3

Aprendizajes esperados

Al final de esta sesión se espera que puedas:

- Crear programas en *MakeCode* que incluyan bucles finitos.
- Crear programas en *MakeCode* que incluyan bucles condicionales del tipo mientras.
- Programar salidas diferenciadas según condiciones de entrada específicas.

Material por grupo

- ○ Anexos 3.1, 3.2 y 3.3
- ○ Acceso a *MakeCode*

Duración sugerida

15% 70% 15%

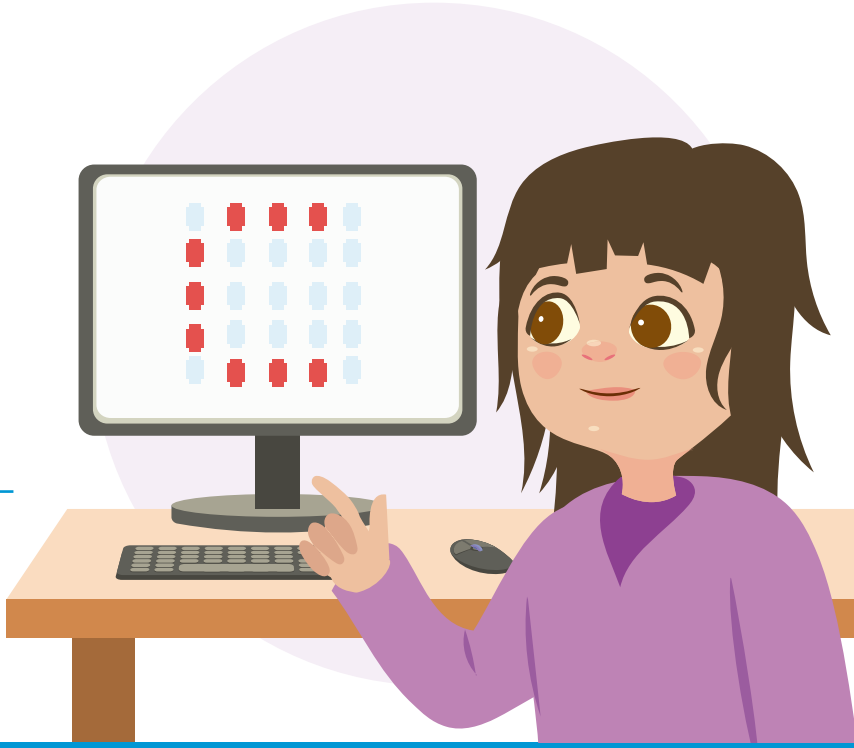
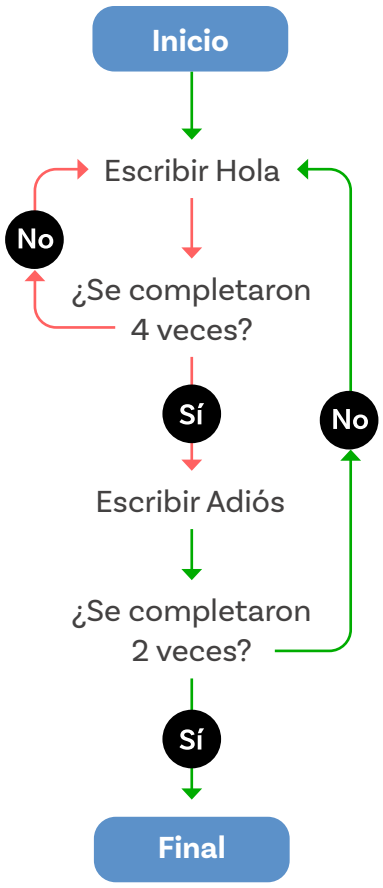


Figura 1. Algoritmo con un bucle dentro de otro bucle



Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

En las sesiones anteriores has podido revisar cómo los bucles o ciclos te permiten simplificar algoritmos cuando hay instrucciones que se repiten más de una vez. Además, practicaste la lectura y creación de diagramas de flujo que incluían tanto condicionales como bucles. Con esta sesión aprenderás que algunos programas usan bucles adentro de otros bucles. Sigue el algoritmo en la columna de la izquierda y responde las preguntas:

¿Cuántas veces escribiste Hola?

¿Cuántas veces escribiste Adiós?

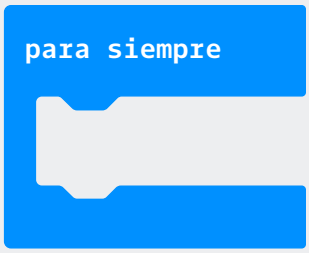
Como aprendiste en la sesión anterior, una de las habilidades del pensamiento computacional es la de reconocer patrones, es decir, identificar características similares o secuencias que se parecen entre sí. En la programación, esta habilidad es muy útil porque permite identificar partes de los programas que se pueden agrupar en un bucle.

Ahora vas a aplicar con la *micro:bit* lo que has aprendido sobre bucles y verás cómo los programas se vuelven más cortos y fáciles de entender. Lee el recuadro Los bucles en *MakeCode* para que sepas cuáles bloques utilizar para crear programas que incorporen ciclos o bucles.

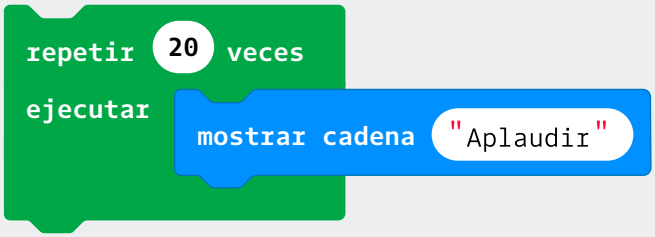
En la siguiente sección vas a realizar algunas actividades que te ayudarán a hacer uso de estos bloques.

Los bucles en *MakeCode*

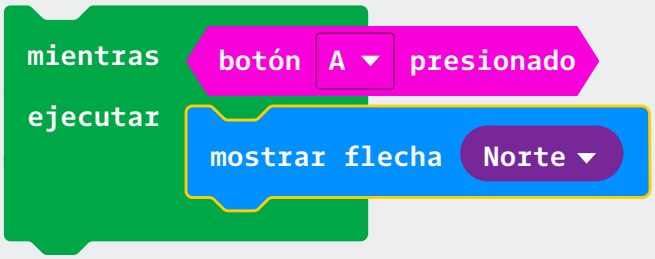
Los programas en *MakeCode* pueden estar en un bucle infinito como este:



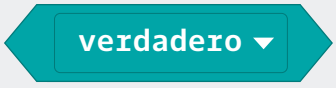
También se pueden programar bucles que se repiten un número de veces. ¿Qué crees que hace el código? ¿Qué pasa si cambias el 20 por un 2?



También, hay bucles que se repiten mientras una condición sea cierta. ¿Qué crees que mostrará el siguiente código?



Ya te habrás dado cuenta de que las condiciones que son ciertas o falsas tienen esta forma:



Enlace

Accede al editor *MakeCode* haciendo clic sobre el código QR o escaneándolo.

Manos a la obra

Conectadas

Esta sección corresponde al 85% de avance de la sesión

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

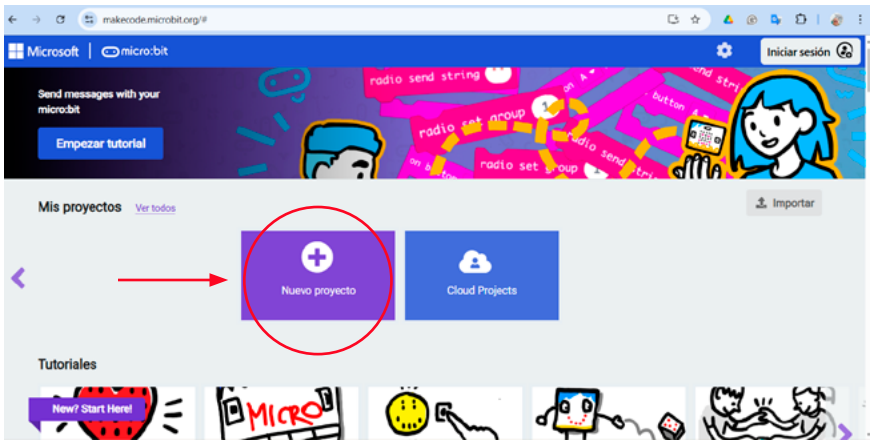
Lean la siguiente situación.

La profesora de tecnología e informática de grado 5 en otra sede les pidió a sus estudiantes que usaran sus micro:bit para presentarse el primer día de clase. Todas las y los estudiantes debían crear una animación que mostrara la primera letra de su nombre, la primera letra de su apellido y su número favorito. La animación debía mostrarse cuatro veces.

La estudiante Cristina Rodríguez hizo el código de la Figura 2 como respuesta a lo que pedía la profesora. Aunque la animación funciona según lo esperado, su profesora cree que el código puede mejorar.

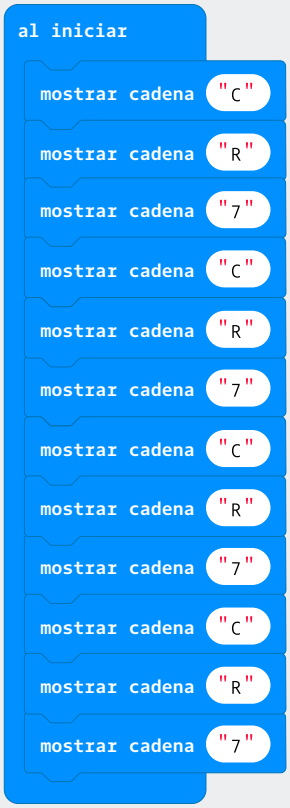
Su misión será programar el código de Cristina, y luego modificarlo haciendo uso de bucles.

1. Ingresen a *MakeCode* y creen un nuevo proyecto.



Recuerda que 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.

Figura 2. Código de Cristina Rodríguez



2 En la caja de herramientas elijan los bloques básicos y ubiquen el de mostrar cadena, y duplíquenlo según lo requieran, para replicar el código de Cristina ver *Figura 2*.

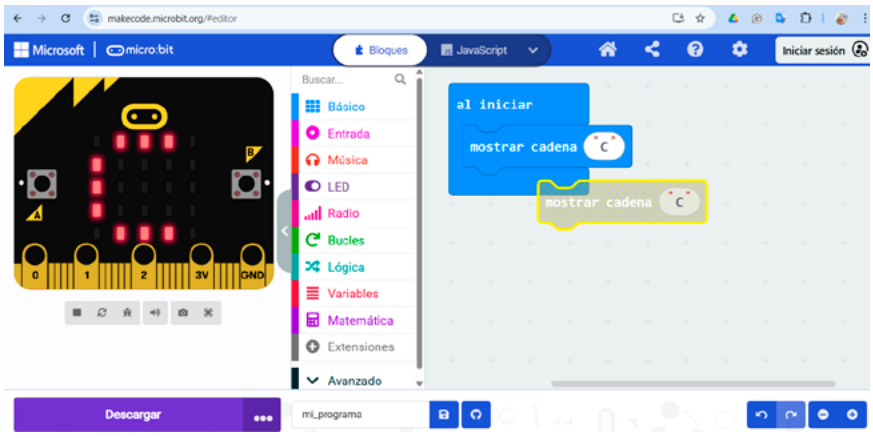
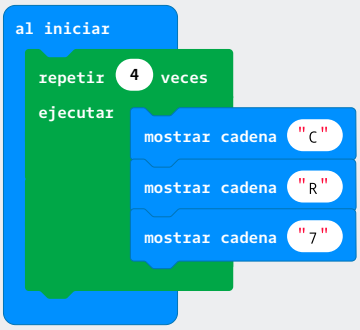


Figura 3. Código de Cristina Rodriguez simplificado con un bucle repetir



3. Cuando ya tenga listo el programa el código en *MakeCode*, discutan: ¿Hace este programa lo que se esperaba?

4. Modifiquen el código para que muestre las iniciales del nombre, apellido y número favorito de una persona del grupo.

5. Identifiquen los pasos que se repiten en el programa que modificaron. Después revisen nuevamente el recuadro Los bucles en *MakeCode* y piensen: ¿Qué tipo de bucle podrían usar para simplificar este código?

6. Ahora creen un nuevo proyecto y programen el código de la *Figura 3*, y luego discutan: ¿Funciona? ¿cuál es la diferencia con el código anterior?

7. Modifiquen el código para que corresponda a las iniciales del nombre, apellido y número favorito de otra persona del equipo. ¿Se tardaron el mismo tiempo haciendo los cambios requeridos? ¿Cómo podrían hacer que la animación se repita 30 veces?

Ya que saben utilizar los bloques de bucles, tendrán un par de misiones para resolver haciendo uso de lo aprendido.

Misión 1:
Planeen y programen una animación en la que utilicen al menos dos bucles. ¿Qué tal si crean una animación que muestre el recorrido de una tortuga desde el mar hasta el lugar de la playa donde anidará para poner sus huevos?

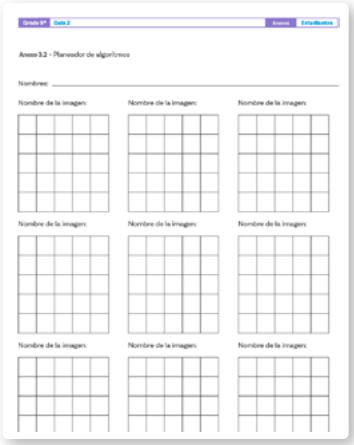
A. Planeen el código por unos minutos. Primero, escriban en sus propias palabras lo que deben hacer.

B. Tengan en cuenta que su animación debe tener al menos dos bucles para cambiar entre imágenes. Pueden usar los bloques mostrar ícono para elegir los íconos prediseñados que prefieran o dibujar los suyos usando el bloque mostrar LEDs.

Anexos

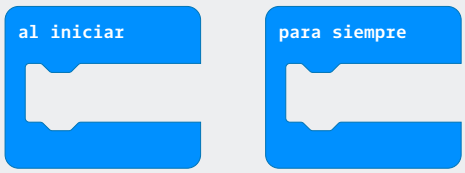
Anexo 3.1

Anexo 3.2



C Usen las plantillas de *Anexo 3.1* y *3.2*, para planear su algoritmo y revisarlo, antes de programarlo en MakeCode.

¿? *¿Qué pasa si cambian el bloque al iniciar, por el bloque para siempre?*



Misión 2:
Van a planear y codificar otra animación, pero esta vez haciendo uso de bucles condicionales, por ejemplo, mediante el uso del bloque Mientras, tal como se ve en la *Figura 4*.

Figura 4. Código con bucle mientras

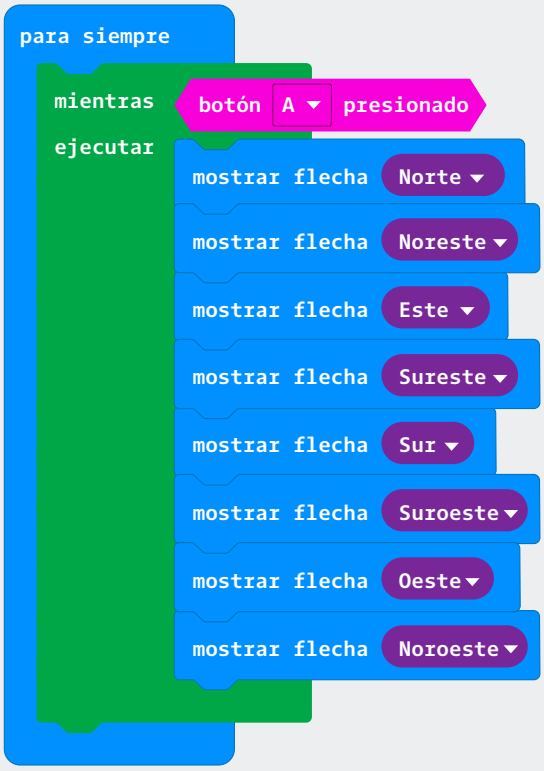
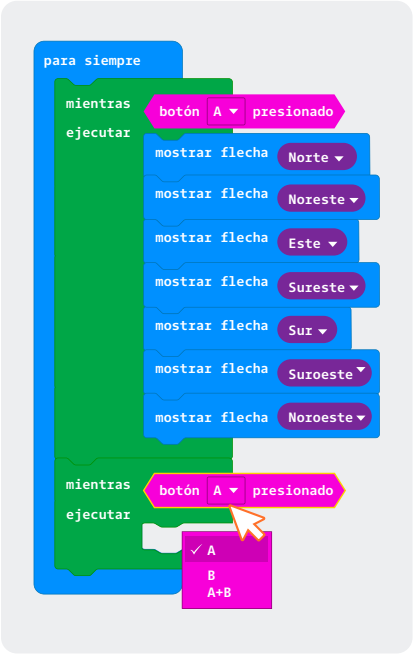


Figura 5. Modificación del código para agregar otro bucle mientras



1. Observen el programa de la *Figura 5*, ¿qué creen que hace?

2. Ahora prográmenlo en el editor *MakeCode* y discutan: ¿Hizo lo que esperaban? Si no se oprime el botón, ¿funciona?

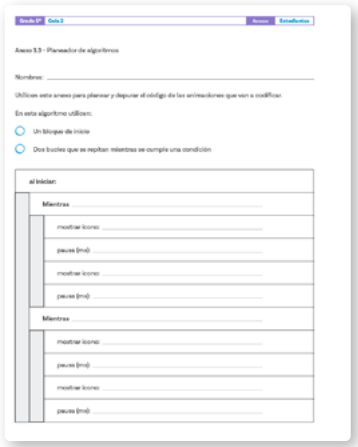
3. Ahora prográmenlo para que, cuando presionen el botón A, haga lo que acaba de hacer y, que cuando presionen B, la flecha gire en dirección contraria. Para realizar este programa usen el mismo bloque para siempre.

Noten que pueden elegir qué botón presionar, haciendo clic en la letra A, como se observa en la *Figura 5*.

Ahora su misión es utilizar el bloque Mientras para diseñar su propia animación. Una vez más, pueden elegir diferentes íconos o dibujar los suyos. Usen las plantillas de los *Anexos 3.2* y *3.3*, para planear su algoritmo y revisarlo, antes de pasar a *MakeCode*. ¿Qué tal si esta vez hacen una animación en la que, al presionar un botón, se muestre a una pequeña tortuga que sale del huevo y hace el recorrido hacia el mar y, al presionar otro botón, se muestre los peligros que debe esquivar mientras va por la playa?

Anexo

Anexo 3.3



Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

Trabaja con una compañera o compañero. Van a jugar un juego de memoria que les ayudará a revisar lo aprendido.

Escriban unas 10 palabras relacionadas con lo aprendido en estas sesiones. Incluyan en su listado las siguientes:

finito	condicional
bloque	patrón
depurar	algoritmo

Ahora, escojan mentalmente una palabra, sin decirla en voz alta. Tomen turnos para describirla para que su compañera o compañero la adivine. Si les surge alguna duda durante el ejercicio, revisen sus notas, revisen las guías previas o pregúntenle a su docente. ¿Cómo les fue? ¿Lograron identificar todas las palabras?

Ahora, completen las siguientes frases:

Los bucles sirven para _____

_____

_____

_____

El bloque Repetir _____

_____

_____

_____

Para hacer bucles infinitos _____

_____

_____

_____

El bloque Mientras _____

_____

_____

_____

Sigan las instrucciones de su docente para compartir sus oraciones y corregirlas, si es necesario.

Finalmente, de forma individual, responde las siguientes preguntas para autoevaluar los aprendizajes de esta sesión.

1. ¿Puedes crear programas en *MakeCode* que incluyan bucles finitos?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes crear programas en *MakeCode* que incorporen bucles condicionales del tipo "mientras"?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", revisa nuevamente el trabajo realizado mientras resolvían las misiones. Luego, ingresa nuevamente a la caja de herramientas de MakeCode, y ubica en la sección de Bucles los bloques "Repetir" y "Mientras". En la sección de Entrada, ubica el bloque "botón A presionado". Crea un programa que, usando estos bloques, realice lo siguiente:**

A. Muestre un número cuando se presionen los dos botones A y B a la vez (botón A+B).

B. Muestre tu nombre cuando se presione el botón A.

C. Muestre una animación de al menos 2 íconos diferentes con pausas entre ellos, que se repita 3 veces cuando se presione el botón B.

D. Muestre una carita feliz si no se presiona ningún botón.

Grado 5° Guía 2

TIC

Sesión 4

Aprendizajes esperados

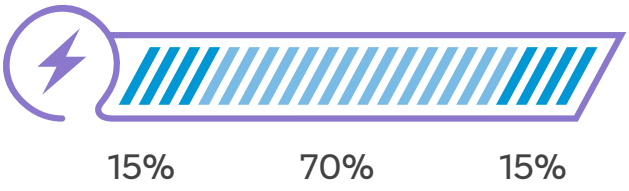
Al final de esta sesión se espera que puedas:

- Utilizar dentro de un programa de *MakeCode* variables de entrada de magnitudes físicas como la temperatura.
- Codificar salidas diferenciadas que respondan a condiciones en valores de entrada dados por el sensor de temperatura.
- Cargar un programa en la micro:bit y verificar su funcionamiento.

Material para la clase

- Acceso a *MakeCode*

Duración sugerida



Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Ahora es el momento de conocer en mayor detalle la *micro:bit*. Ya has explorado el entorno de programación *MakeCode* y viste cómo actúa el dispositivo usando el simulador.

1. Entrar a *MakeCode*.
2. En el frente de la *micro:bit*, identificar los siguientes elementos:
   - Botón A.
   - Botón B.
   - El panel de 25 LED (5x5) de la *micro:bit*.
3. Por el reverso de la *micro:bit* identificar:
   - El procesador.
   - La brújula (*compass*).
   - El acelerómetro.

Figura 1. Vista frontal y posterior de la micro:bit

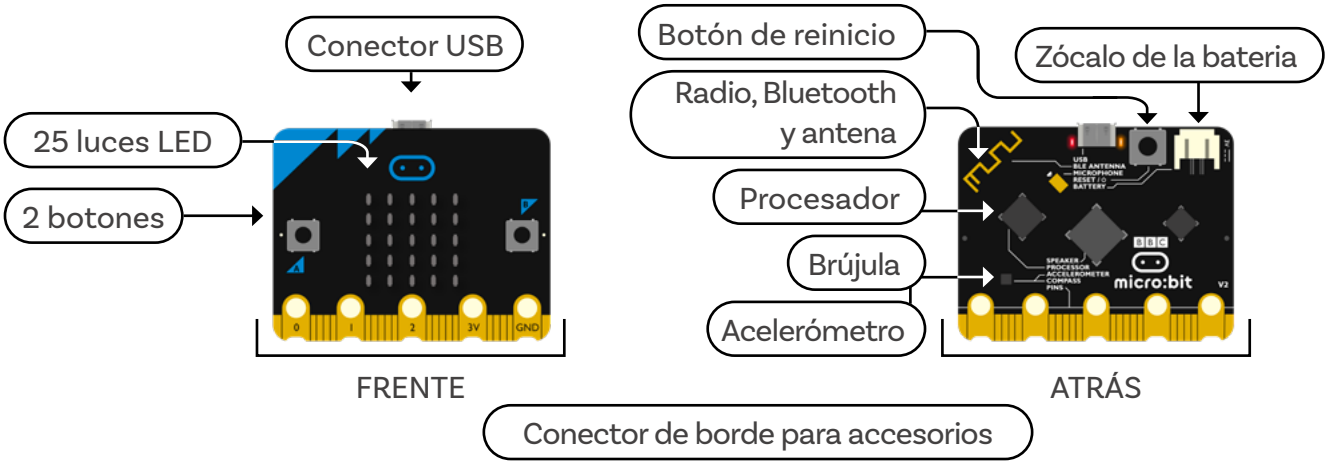
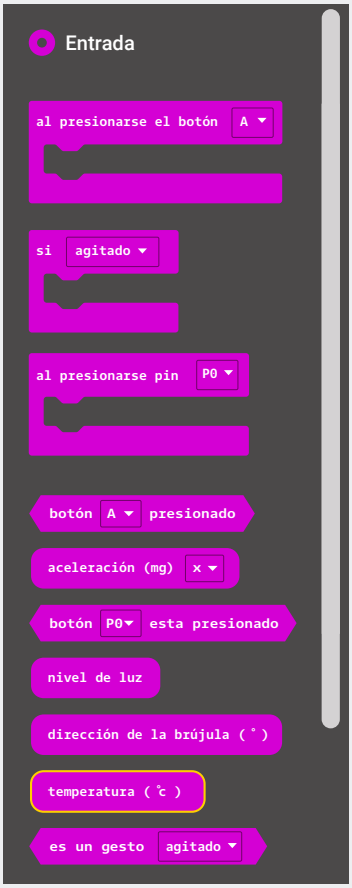


Figura 4. Sensor de temperatura en la caja de herramientas de MakeCode



Los sensores, incluido el de temperatura se pueden acceder desde la caja de herramientas de *MakeCode*, en la sección de Entradas.

Sensores

Un sensor es un dispositivo capaz de captar una variable física como la temperatura.

La *micro:bit* tiene un sensor de temperatura ubicado en el pequeño procesador que tiene (ver *Figura 2*). Con este sensor puedes saber a qué temperatura está la *micro:bit*.

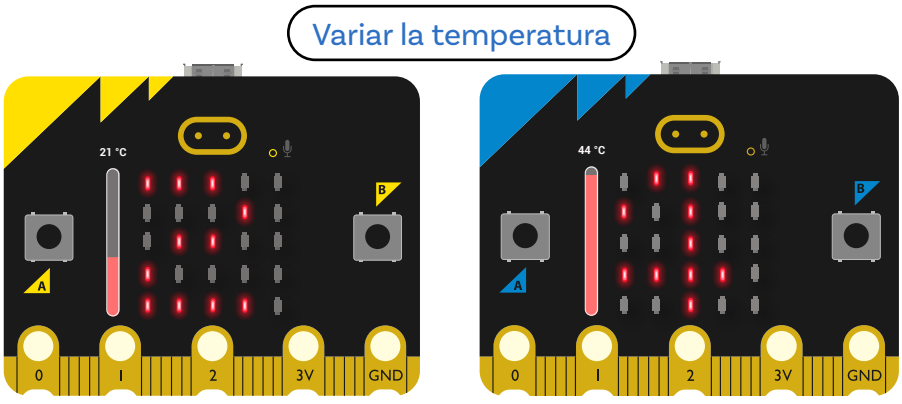
Figura 2. Sensor de temperatura en la micro:bit

La temperatura se mide en °C (grados Celsius), por lo que es una *variable numérica* que asume varios valores.

A los sensores también los llamamos Entradas.

Al utilizar el sensor de temperatura, en el simulador se habilita una imagen de un termómetro que permite variar la temperatura, como se observa en la *Figura 3*.

Figura 3. Herramienta para modificar la temperatura al simular el código



Manos a la obra

Conectadas

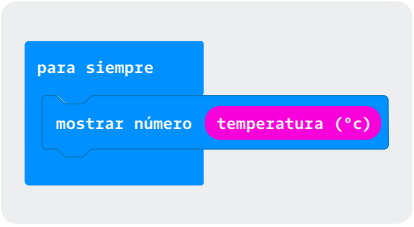
Esta sección corresponde al 85% de avance de la sesión

Antes de empezar con ejercicios de aplicación, vamos a probar el sensor de temperatura. Trabaja en pareja con alguna compañera o compañero según te indique tu docente.

Observen el código de la *Figura 5*. ¿Qué creen que hace? ¿Qué esperan que pase cuando lo ejecuten?

Prográmenlo en *MakeCode* y observen su comportamiento. ¿Es lo que esperaban?

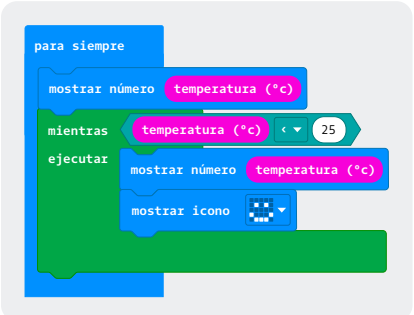
Figura 5. Bucle infinito que muestra lecturas del sensor de temperatura



El código que muestra la *Figura 5* utiliza un bucle infinito (que se repite siempre) y una nueva entrada: la temperatura.

> *¿Les muestra la temperatura en la matriz LED?*
>
> *¿Pueden variar la temperatura en el simulador?*

Figura 6. Código con bucle mientras activado según lectura del sensor de temperatura



Guíense de las imágenes en la *Figura 3* para cambiar el valor del termómetro y simular que aumenta o disminuye la temperatura.

El programa de la *Figura 6* se hizo para ayudar a controlar la temperatura de un salón de clase en una ciudad de clima caliente. Ahora complementen lo que le falta a su programa para tener el código de la *Figura 6*. Fíjense que se hace uso de bloques de entrada, de bucles y básicos. ¿Cuál creen que será el resultado? Pueden modificar el valor en el termómetro para probar su funcionamiento.

Intenten modificar la temperatura para que sea de menos de 25 grados. ¿Pueden ver la carita feliz?

Prueben con una temperatura de 26 o más grados. ¿Pueden ver la carita feliz?

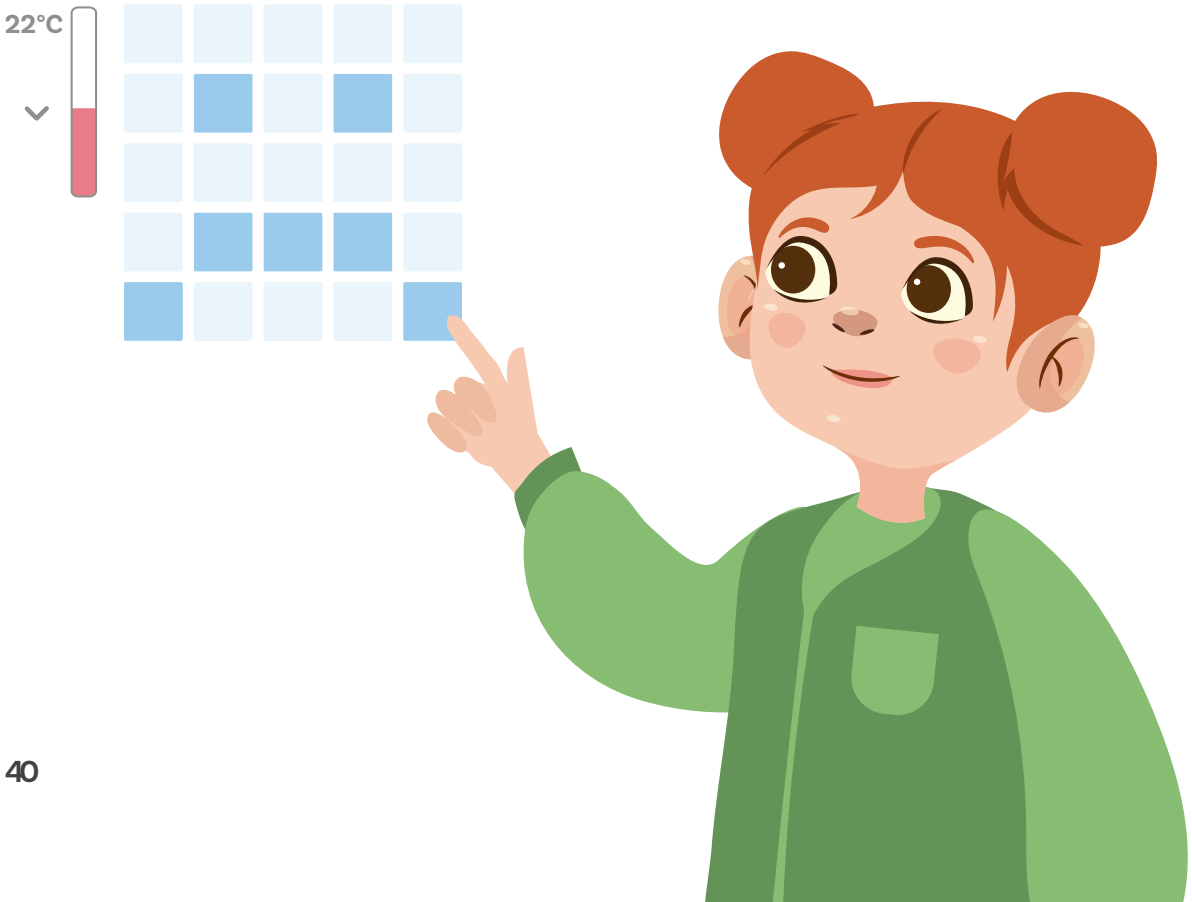
Como lo notaron, la carita solo aparece cuando se cumple la condición que se ha programado. Como la temperatura es una variable numérica, la condición es una comparación entre el valor que detecta el sensor (o el que simulan con el termómetro) y el valor máximo que se ha determinado (25 grados).

Ahora:

1. Complementen este programa haciendo que salga una cara triste a una temperatura menor a 23 grados. Simulen y prueben su código.

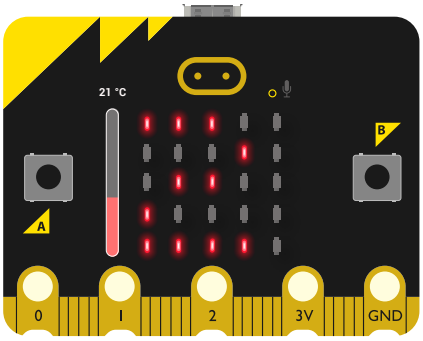
2. Ajústenlo para que cumpla con las siguientes condiciones:

   - A menos de 23 grados deberían ver la cara triste.
   - A más de 25 grados deberían ver la cara feliz y el valor de la temperatura.
   - Para el resto de los casos deberían ver solamente la temperatura.



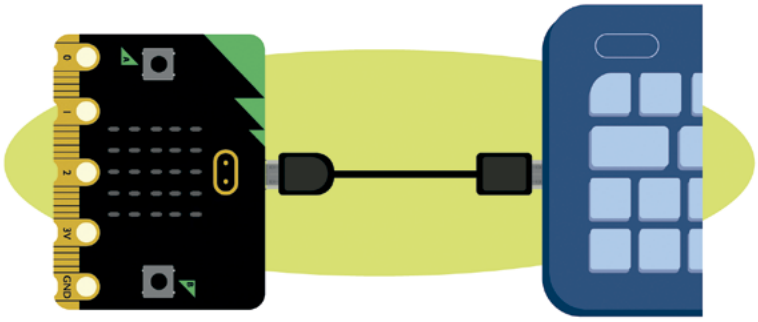
Para ir más lejos

Si tienes una micro:bit a tu alcance es el momento de probar tu programa.

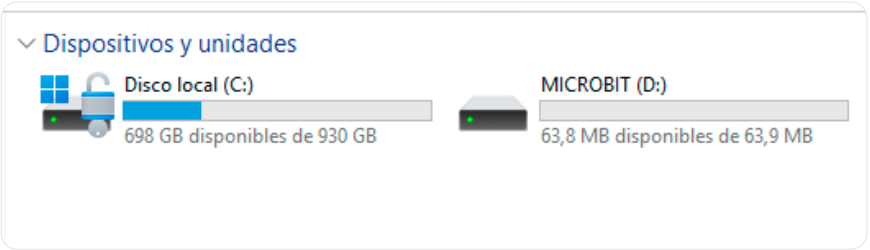


Ya han trabajado con el simulador. Si cuentan con una *micro:bit* podrán transferir el programa y probarlo igualmente en la tarjeta. Para ello:

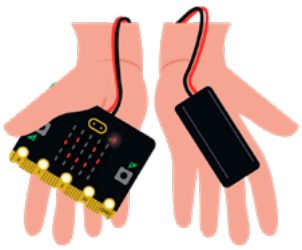
1. Conecten la *micro:bit* utilizando el cable USB a un puerto USB de su computador.



2. La *micro:bit* se encenderá y en su computador aparecerá como si tuvieran un disco duro adicional conectado.

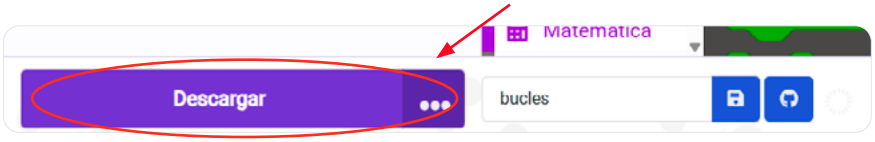


3 La caja de baterías solo la necesitarán cuando quieran que la *micro:bit* funcione sin conexión al computador.

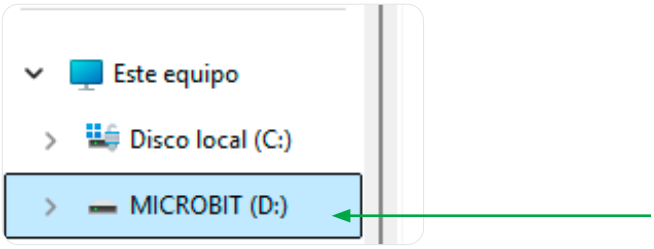


Si programas desde un dispositivo celular o una tableta también puedes descargar tu programa en la micro:bit. Pide ayuda a tu docente para hacerlo.

4 Utilicen el botón Descargar de la parte baja del editor, guarden el archivo con el nombre que deseen. Notarán que se descarga con la extensión .hex

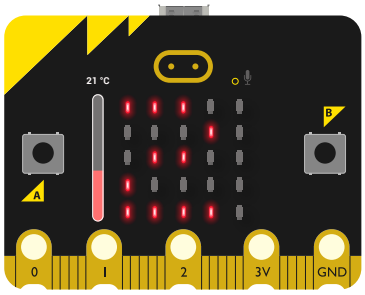


5 Arrastren el archivo en la *micro:bit* que aparece como si fuera un disco o una memoria externa llamada *MICROBIT*.



6 Si conectan la batería, podrán desconectar la *micro:bit* sin perder el programa que hayan cargado en ella.

7 Es su turno, ¿funciona? La *micro:bit* puede ahora funcionar autónomamente sin conexión al computador.



Prueba tu programa en la *micro:bit*. Recuerda que debes grabar el programa en una ubicación del computador y luego transferirlo vía USB al dispositivo. Prueba tu programa usando tus manos para calentar la *micro:bit*.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

Ya han aprendido que la *micro:bit* se puede programar para detectar variables físicas, por ejemplo, por medio de su sensor de temperatura, y así poder asegurar que se mantiene un clima adecuado en un espacio como el salón de clase. Discutan como equipo la forma en que la lectura y comparación de valores de temperatura puede utilizarse para crear otros programas. ¿Qué se les ocurre que podrían hacer con esto que han aprendido? ¿Podría aplicarse con relación a temas de salud? ¿En algo relacionado con la agricultura? ¿Podría tener utilidad en la industria? ¿Podría servir para crear programas que sirvan para apoyar la exploración de las profundidades del océano o del espacio?

Anoten sus ideas.

Hagan un dibujo para ilustrar una de las situaciones que imaginaron previamente, en las que se hace uso de un programa que mide la temperatura de algo y según el resultado permite prender o apagar una máquina, mostrar una alerta o realizar alguna otra acción automatizada.

Por último, piensen... ¿cómo podría utilizarse lo aprendido para ayudar a salvar a las tortugas? Eso es algo que aprenderán en la siguiente sesión.

Ahora, de forma individual, responde estas preguntas:

1. ¿Puedes utilizar dentro de los programas de *MakeCode* variables de entrada de magnitudes físicas como la temperatura?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes codificar salidas diferenciadas que respondan a condiciones en valores de entrada dados por el sensor de temperatura?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

3. ¿Puedes cargar un programa en la micro:bit y verificar su funcionamiento?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

> **Si tus respuestas a las preguntas anteriores fueron "Parcialmente" o "Aún no", no dudes en regresar y ensayar por tu cuenta las instrucciones de la sección Manos a la obra. Consulta con tu docente las dificultades que tengas o las dudas que te surjan.**

Cuando ya hayas aclarado tus dudas, resuelve el siguiente reto:

Te han pedido programar la *micro:bit* para simular el control de temperatura de la incubadora de huevos de gallina que hay en la finca de una amiga de tu familia. Para que los pollitos no se afecten, los huevos deben mantenerse entre los 37 y 38 grados de temperatura. Por tanto, mientras la temperatura sea mayor a 36 grados o menor a 39, tu programa debe mostrar una carita feliz. En los demás casos, debe mostrar una carita triste. Cuando tengas listo el reto, muéstrale el código a tu docente.

Grado 5º | Guía 2

TIC

Sesión 5

Aprendizajes esperados

Al final de esta sesión se espera que puedas:

- Diseñar el diagrama de flujo de un algoritmo con bucles condicionales.
- Crear un programa que ejecute tomas de decisión basadas en la medición del sensor de temperatura.

Duración sugerida

15% 70% 15%

Material para la clase

- Acceso a *MakeCode*

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

¿Alguna vez te has preguntado cómo la programación puede ayudar en la vida cotidiana? Hoy existen muchísimas soluciones basadas en tecnología que ayudan a facilitar y mejorar la vida y el trabajo a las personas.

Imagina que quieres cuidar mejor de la naturaleza y proteger a los árboles. ¿Sabías que se puede usar la programación para ayudar con eso? Por ejemplo, se puede crear un programa que monitoree la cantidad de agua que reciben los árboles en un parque. Este programa podría enviar alertas a las y los cuidadores del parque cuando los árboles necesiten más agua y ayudarles a mantenerlos saludables.

Además, la programación también puede ayudar a hacer las cosas más fáciles y rápidas. ¿Has escuchado sobre la automatización de tareas? Es como si se le enseñara a una computadora a hacer cosas por las personas. Por ejemplo, se puede programar un sistema que encienda y apague las luces o que prenda automáticamente el calentador de agua a la hora que se quiera tomar un café o una aromática.

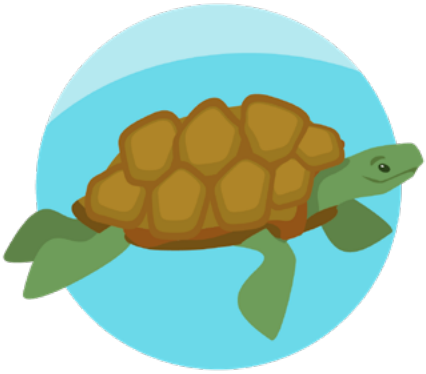
En la siguiente actividad vas a trabajar con un problema de la vida real donde verás cómo con ayuda de la computación se puede, incluso, ayudar a salvar a las tortugas.

Manos a la obra

Aplicando lo aprendido

Esta sección corresponde al 85% de avance de la sesión

Muchas especies de tortugas marinas están en peligro de extinción. Por eso, las y los biólogos de la conservación y muchas personas voluntarias recogen cada año los huevos que depositan las tortugas en las playas y los llevan a incubadoras para protegerlos de depredadores y humanos. Los huevos de tortuga, como los de todos los reptiles, son muy sensibles a la temperatura y si se exponen a más de 34°C no se desarrollan. De la misma manera, temperaturas inferiores a 26°C no permiten que crezcan los embriones.



Un centro de preservación de tortugas marinas ha pedido ayuda, para programar un dispositivo que les permita mantenerse informados sobre la temperatura del sitio de incubación. Quieren saber si es muy baja, adecuada o muy alta para el desarrollo de los huevos.

Trabaja con una compañera o compañero. Su misión será programar la *micro:bit* para que detecte la temperatura e informe el valor en el tablero de LED. Además, deberá avisar a las y los biólogos encargados cuando la temperatura sea muy baja con un mensaje que diga “T. baja”, cuando la temperatura sea normal “T. normal” y cuando la temperatura sea muy alta, “T. alta”. Si quieren pueden inventar íconos que reemplacen los textos pero que resulten evidentes para quien observa.

Para empezar, hagan un diagrama de flujo que indique lo que debe hacer el programa. Luego, usen *MakeCode* para escribir estas instrucciones en el lenguaje de bloque y utilicen el simulador para ver si su programa funciona apropiadamente.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

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

1. ¿Puedes diseñar el diagrama de flujo de un algoritmo con bucles condicionales?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

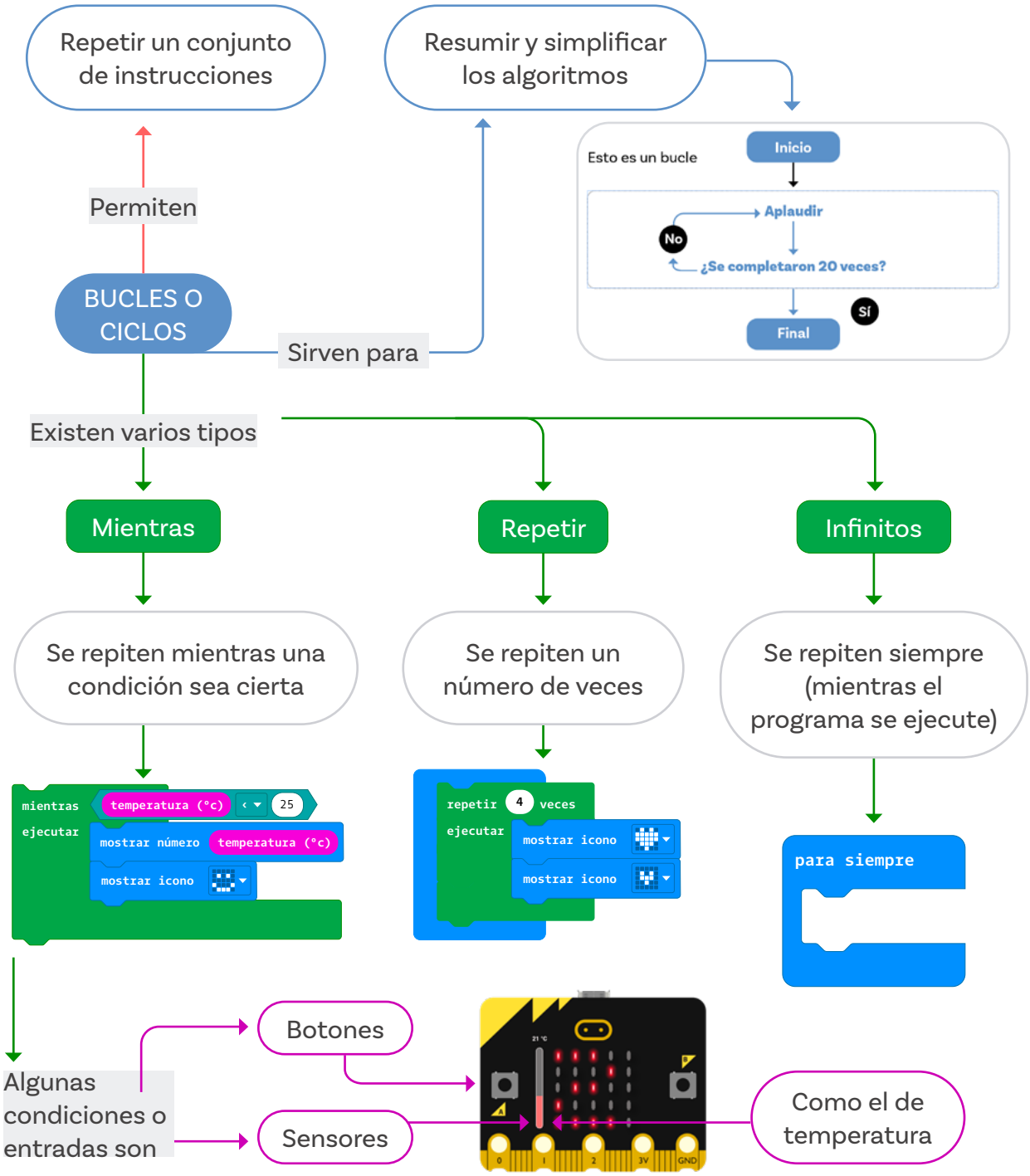
2. ¿Puedes crear un programa que ejecute tomas de decisión basadas en la medición del sensor de temperatura?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

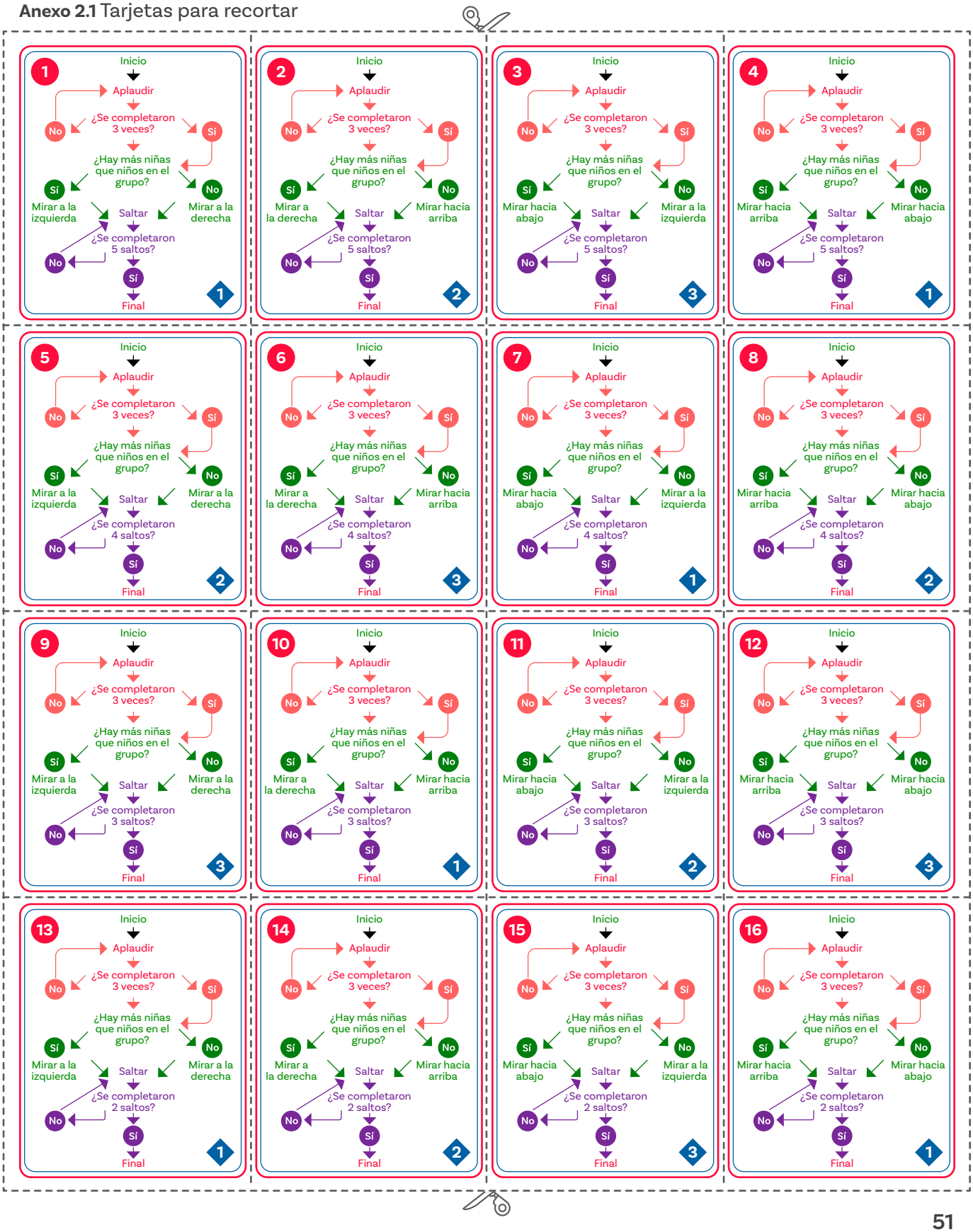
Si pudiste resolver la misión planteada en esta sesión, seguro que respondiste afirmativamente las preguntas anteriores. En caso de que no haya sido así, verifica con tu compañera o compañero y docente lo que no haya podido resultarles y asegúrate de aclarar dudas.

El siguiente es un gráfico de anclaje que condensa lo aprendido en las sesiones de esta guía. Compleméntalo con ejemplos y dibujos que te permitan recordar mejor lo aprendido.

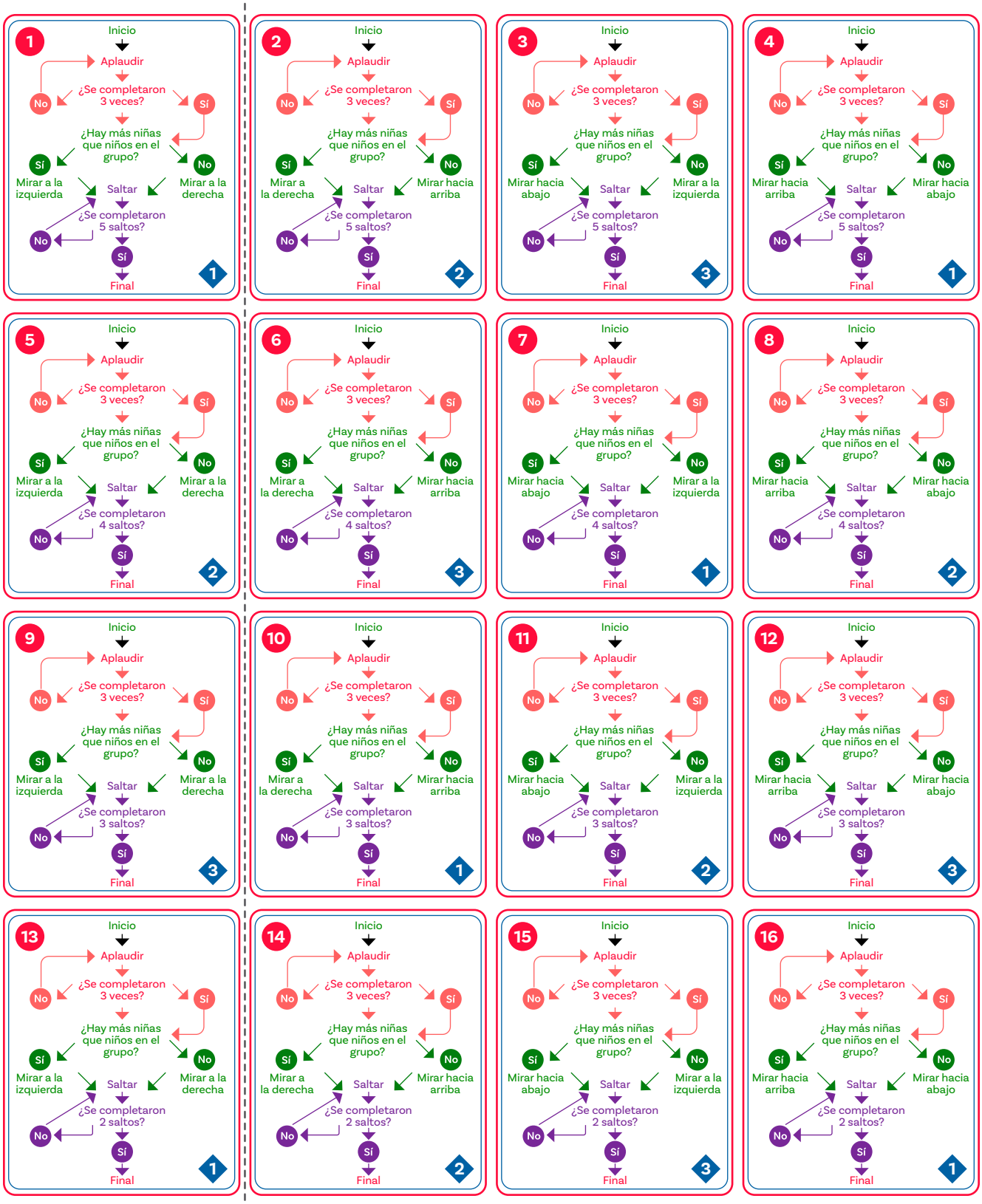
Figura 1. Gráfico de anclaje sesión 5



Anexo 2.1 Tarjetas para recortar



Anexo 2.2 Tarjetas para encontrar la carta



Anexo 3.1 Planeador de algoritmos

Nombres: _____

Utilicen este anexo para planear y depurar el código de las animaciones que van a codificar.

En este algoritmo utilicen:

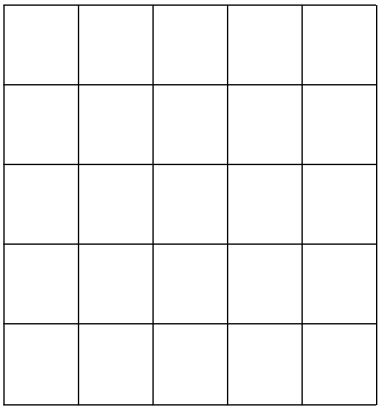
- ◯ Un bloque de inicio.
- ◯ Dos bucles que se repitan un número exacto de veces.

al iniciar:
repetir _____ veces:
mostrar ícono: _____
pausa (ms): _____
mostrar ícono: _____
pausa (ms): _____
repetir _____ veces:
mostrar ícono: _____
pausa (ms): _____
mostrar ícono: _____
pausa (ms): _____

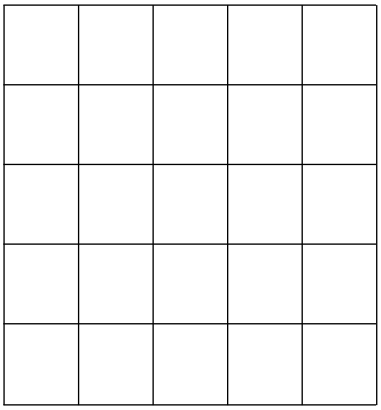
Anexo 3.2 Planeador de algoritmos

Nombres: _____

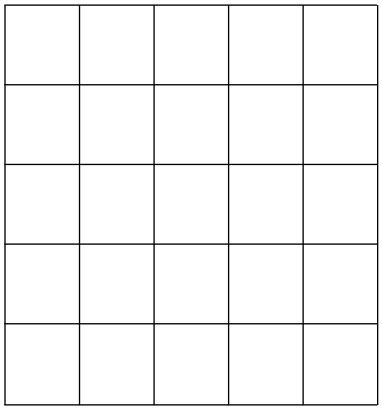
Nombre de la imagen:



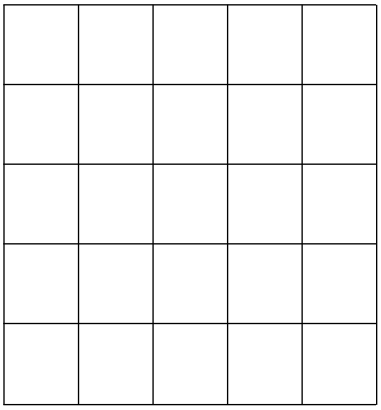
Nombre de la imagen:



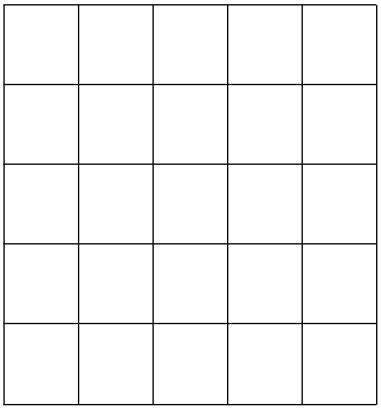
Nombre de la imagen:



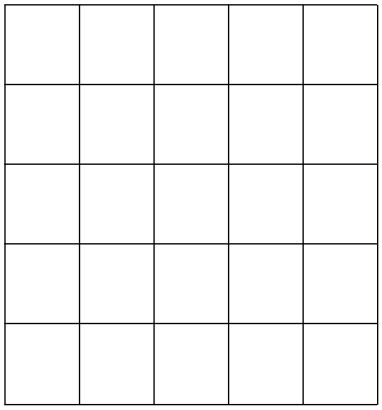
Nombre de la imagen:



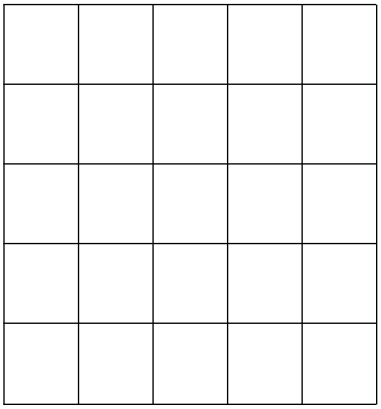
Nombre de la imagen:



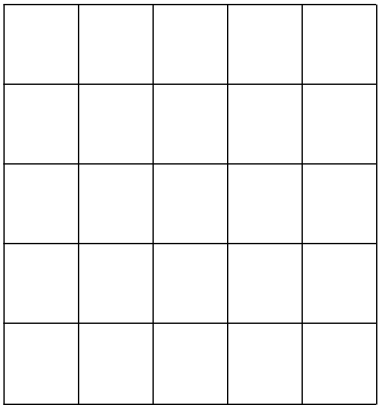
Nombre de la imagen:



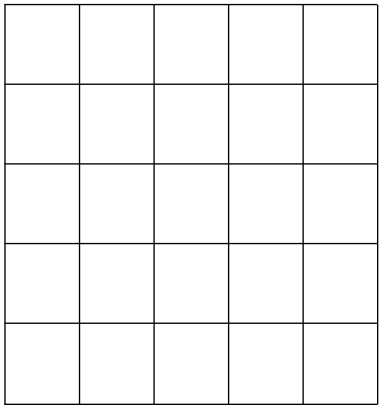
Nombre de la imagen:



Nombre de la imagen:



Nombre de la imagen:



Anexo 3.3 Planeador de algoritmos

Nombres: _____

Utilicen este anexo para planear y depurar el código de las animaciones que van a codificar.

En este algoritmo utilicen:

- ◯ Un bloque de inicio.
- ◯ Dos bucles que se repitan mientras se cumple una condición.

al iniciar:
Mientras _____
mostrar ícono: _____
pausa (ms): _____
mostrar ícono: _____
pausa (ms): _____
Mientras _____
mostrar ícono: _____
pausa (ms): _____
mostrar ícono: _____
pausa (ms): _____

TIC

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}