



# FICHA 2

## SALVANDO A LAS TORTUGAS

Sesión 1



### Aprendizajes

Al final de esta actividad se espera que puedas:

- Utilizar **condicionales** para decidir realizar o no una acción.
- Utilizar **condicionales** para controlar la repetición de un conjunto de acciones.
- Interpretar y hacer **diagramas de flujo** sencillos.
- Utilizar **variables de entrada** de magnitudes físicas como la **temperatura**.
- Mostrar una variable numérica, como la temperatura, en el arreglo de LED.
- Cargar un programa en la **micro:bit** y verificar su funcionamiento.



### Lo que sabemos, lo que debemos saber



En las actividades pasadas aprendiste a describir un proceso paso a paso y a usar un lenguaje para comunicar este proceso a un **procesador**. Utilizaste un lenguaje de flechas para dar instrucciones a un **procesador** y usaste algunos comandos de un lenguaje de bloques de la **micro:bit** en el editor y simulador **MakeCode**.

Existen diferentes códigos o lenguajes que sirven para programar distintos **procesadores**, pero todos describen una secuencia lógica de pasos o instrucciones. Algunas acciones requieren pocos pasos, como mostrar el corazón en el tablero de LED de la **micro:bit**, mientras otras pueden requerir miles y miles de pasos, como por ejemplo calcular la ruta más rápida para llegar de un punto a otro en el mapa de una ciudad.

Quienes programan usan **algoritmos** expresados en **diagramas de flujo** para representar procesos que tienen muchas instrucciones. Así se pueden organizar las instrucciones en el orden adecuado para luego realizar correctamente la **programación**.

Muchas actividades que se pueden expresar mediante un conjunto de pasos ordenados se pueden también expresar en un **diagrama de flujo**. Examina el siguiente caso para comprender qué es un **diagrama de flujo**:

En la rectoría del colegio Las Palmas se tiene un archivador donde se encuentran las carpetas con los documentos del estudiantado.

Cada estudiante tiene asignada una carpeta donde se guardan sus documentos, entre los que se encuentra una ficha médica que indica alergias y contacto de acudientes.

El día de ayer se adelantó el proceso de archivar todas las fichas médicas del estudiantado, cada una en su carpeta, pero hoy no se encontró la ficha de la estudiante Juana Suárez.

La explicación más probable es que la ficha se haya archivado en alguna otra carpeta quedando pegada detrás de la ficha de otra estudiante.

Es decir que la ficha médica de Juana Suárez puede estar en cualquiera de las carpetas de las más de 600 estudiantes de la institución educativa.



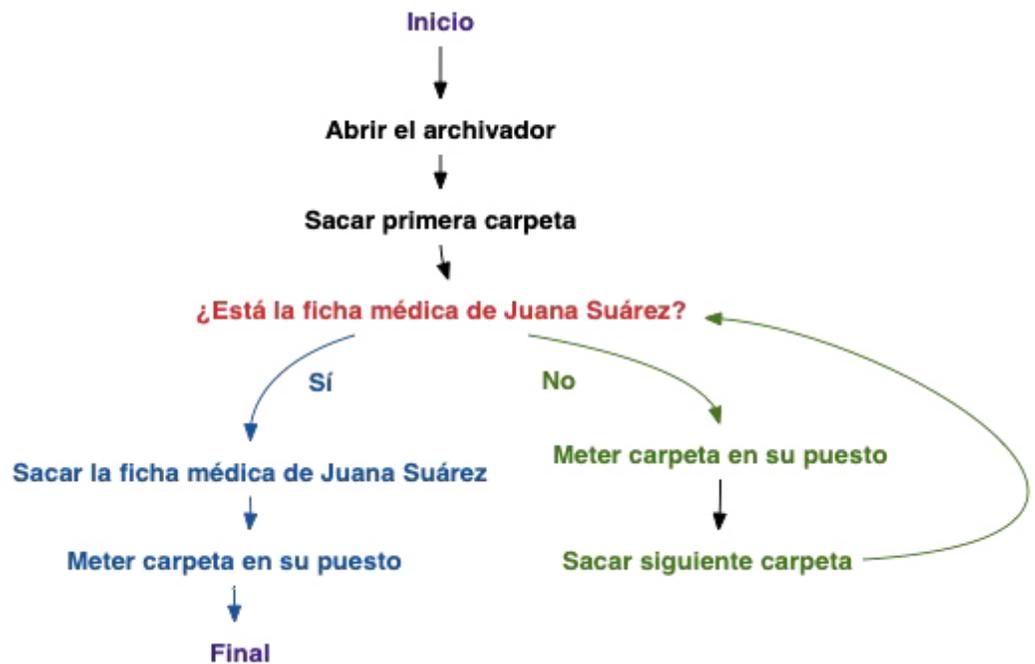


El proceso por realizar implica buscar en todas las carpetas.

A una persona se le diría simplemente que vaya mirando cada carpeta en búsqueda de la ficha y que cuando la encuentre la archive en la carpeta correcta.

Pero a un procesador se le deben dar las instrucciones paso a paso, por lo cual esta instrucción se debe detallar más.

Examina el diagrama de flujo que se encuentra a continuación y verifica que las instrucciones son detalladas y correctas.



En este diagrama podrás observar que hay un **condicional** en **rojo**. En un condicional si la condición es **verdadera (sí)**, se realizan unos pasos, pero si es **falsa (no)** se realizan otros pasos. En nuestro ejemplo el **condicional** tiene la pregunta **¿Está la ficha médica de Juana Suárez?**

Según la respuesta se toma uno de los dos caminos: si no se encuentra la **ficha** de Juana Suárez seguimos el camino **verde** que al final nos regresa de nuevo al **condicional**. Para encontrar la ficha de Juana se repetirá este bloque de instrucciones verdes hasta encontrarla.

Cuando el condicional sea cierto, esto es encontramos la ficha de **Juana Suárez**, seguimos el camino **azul** del diagrama de flujo.



El rector observó este diagrama de flujo e indicó que le parece que las instrucciones no están completas. Por ejemplo, ¿qué pasa si se llega a la última carpeta y no se encuentra la ficha? Sigue el diagrama de flujo anterior e indica qué sucedería si esto llegase a pasar.

El rector también indica que debería archivar de una vez la ficha médica de Juana Suárez en el lugar apropiado, pero el diagrama de flujo no parece decir que se haga. También le gustaría que quedara cerrado el archivador al concluir el proceso.

Piensa tú o con tu grupo sobre los cambios que deberán realizarse en el diagrama de flujo anterior para resolver estos dos problemas.

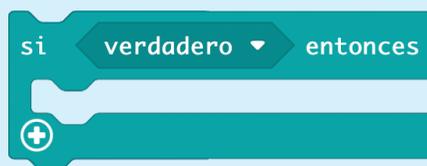
En el ejemplo que vimos, si se llega a la última carpeta y la ficha no se encuentra, lo que sucedería es impredecible si fuese un computador quien hace la búsqueda, dado que esta situación no está contemplada. Una razón por la que fallan algunos programas de computador es cuando se presenta una condición que no contemplaron quienes diseñaron el programa. Los computadores simplemente siguen instrucciones de modo que cuando se presenta una situación no contemplada el resultado puede ser catastrófico. Encontrar este tipo de error se denomina **depurar** el algoritmo.

En esta situación se uso (en verde en el diagrama de flujo) un condicional para controlar la repetición de 2 instrucciones. Lee la siguiente información, te servirá para la sección **conectadas**.

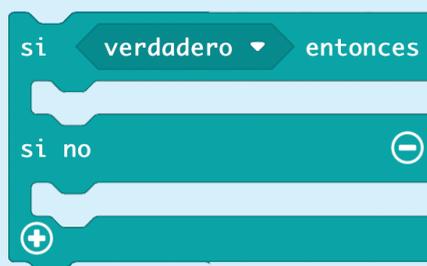
**EN RESUMEN:**

Un **diagrama de flujo** se compone de instrucciones, condicionales y flechas que permiten describir el orden en que se deberán ejecutar las instrucciones.

Un **condicional** permite realizar o no un conjunto de instrucciones o seleccionar entre dos alternativas de instrucciones según si se cumple la condición o no.



Todas las acciones que están dentro de este espacio se ejecutarán si la condición se cumple, es decir, es VERDADERA



Todas las acciones que están dentro de este espacio se ejecutarán si la condición no se cumple, es decir, es FALSA





## Desconectadas

Es el momento de aplicar lo que hemos aprendido en la sección anterior. Vamos a jugar con unas cartas poco usuales; cada una describe una coreografía en forma de diagrama de flujo. Las instrucciones del juego son las siguientes:

1. Imprimir el anexo 1 y recortar las cartas.
2. Imprimir una copia del anexo 2 para cada grupo que participará en el juego (ver el siguiente paso para determinar el número de grupos).
3. Organizarse en grupos, idealmente de 3 a 5 integrantes. Sin embargo, en caso de pocos participantes, se puede jugar con grupos unipersonales.
4. Entregar a cada grupo una ficha resumen de todas las coreografías (anexo 2).
5. Colocar todas las cartas sobre la mesa de modo que no se puedan ver los diagramas de flujo. Cada grupo toma una tarjeta que no puede ver el resto de grupos.
6. Cada grupo designa a una persona que tenga buena memoria, dado que deberá pasar al frente a realizar la coreografía descrita en la carta seleccionada.
7. Cuando pasa al frente una persona y hace la coreografía, los otros grupos tendrán 1 minuto para buscar entre todos los diagramas de flujo cuál es el correcto y escribir su número (se encuentra en la esquina superior izquierda de la tarjeta, dentro de un círculo) en un papel. El segundo condicional en el algoritmo, en verde en la tarjeta, se refiere al grupo que presenta la coreografía.
8. A la señal de quien orienta el juego se muestran todas las respuestas. Si alguien habla en voz alta dando la respuesta antes de esta señal, pierde 2 puntos.
9. Se asignarán los puntos indicados en la carta (en la esquina inferior derecha de la tarjeta) a quien o quienes hayan encontrado la solución. El grupo que hizo la representación también recibe ese puntaje. Si ningún grupo encuentra el diagrama correcto no se dan puntos.



Si la actividad es en familia se puede jugar individualmente como se propuso para cuando son pocas personas.

**Actividad desconectada complementaria:** Al final de la ficha se presenta una segunda actividad desconectada denominada **Juego de lógica y condiciones** en el cual se podrá profundizar en expresiones y condiciones lógicas y su valoración.

### EN RESUMEN:

Hemos visto que los **condicionales** sirven en dos situaciones:

- Cuando se quiere decidir qué instrucción se ejecutará de acuerdo con una condición.
- Cuando se quiere controlar la ejecución repetida de una o varias instrucciones.



Hacer diagramas de flujo que incluyen condicionales y repetición de algunas instrucciones es muy útil para resolver problemas con computación. Dado que no acostumbramos describir procesos de esta manera, te invitamos a seguir practicando con la siguiente situación.



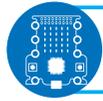
### Un café con leche que no es café con leche

Ha llegado una nueva máquina para preparar diferentes tipos de bebidas a base de café. Entre sus posibilidades está preparar bebidas con leche como el capuchino, las cuales requieren leche “cremada”. Tendrás la tarea de proponer un **algoritmo** en un **diagrama de flujo** con el que posteriormente se pueda escribir un **programa** en un **procesador** de esta máquina el cual estará a cargo de “cremar” la leche. Probablemente sea conveniente que utilices **condicionales** que permiten controlar la ejecución repetida de algunas instrucciones.

Una vez termines, busca otra persona o tu docente para que haga la labor del **depurador**. No queremos que la máquina funcione mal.

En una entrevista con una barista, que es como se llaman las personas expertas en preparar diferentes bebidas a base de café, se logró la siguiente información:

1. La clave de un buen capuchino es cremar la leche de forma apropiada. Las máquinas que hacen capuchino tienen un dispositivo que inyecta vapor de agua a la leche, calentándola y generando una espuma estable y de sabor agradable.
2. El proceso de cremar la leche es muy delicado, porque se deben tener en cuenta varios aspectos:
  - Para preparar una taza de capuchino se requieren 300 ml de espuma. Si el proceso de cremado se realiza bien, el volumen de espuma que se obtiene es el doble del volumen de leche fría que se alista para “cremar”.
  - La leche logra su mejor punto de “cremado” entre 60 y 63 grados centígrados. Si se calienta más que esto se pone amarga y si se calienta menos no se forma una espuma estable
  - Terminado el proceso se debe verter la leche “cremada” sobre un café negro.



### Conectadas: manos a la micro:bit

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

Recuerda que para trabajar con la **micro:bit** necesitarás:

1. Entrar a **MakeCode** en tu computador o al editor en línea si tienes internet.
2. Identificar en la **micro:bit** los siguientes elementos:
  1. Botón A
  2. Botón B
  3. El arreglo de 25 LED (5x5) de la **micro:bit**

1. El procesador
2. La brújula (compass)
3. El acelerómetro

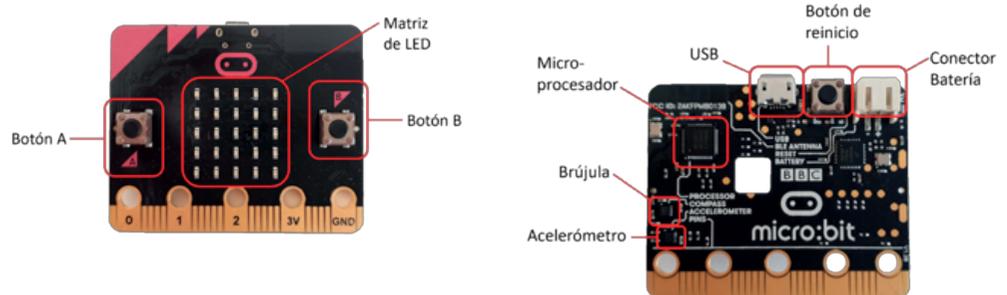


Cuando se tiene un bloque con una condición aparece en una forma como esta:



La condición puede dar como resultado verdadero o falso.

En el ejemplo de arriba, se mostrará la flecha mientras el botón A esté presionado.



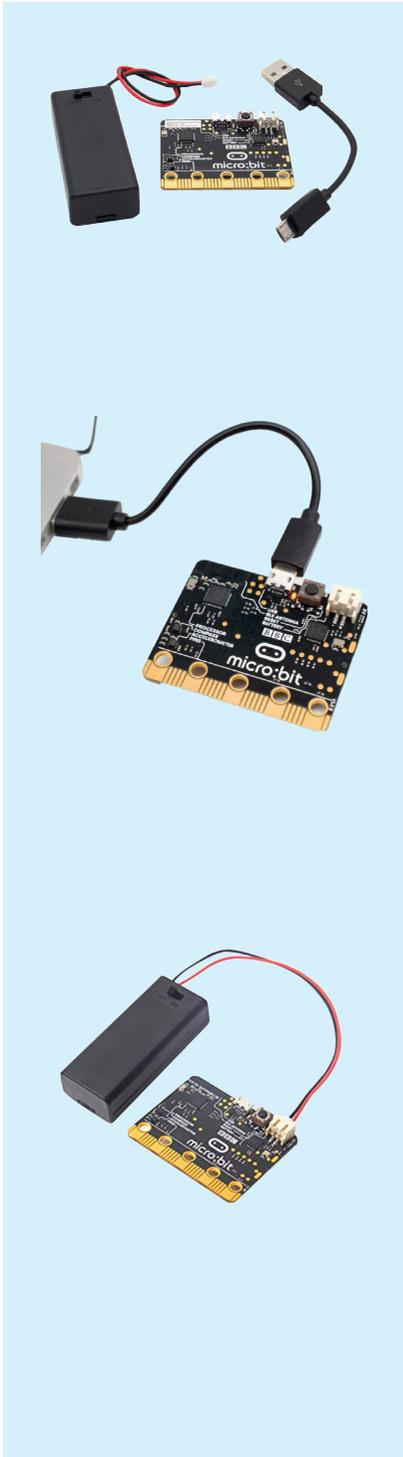
Es hora de programar y simular en el editor **MakeCode** un programa para verificar si funciona.

1. Este **programa** que se encuentra a la derecha, ¿qué crees que hace?
2. Ahora prográmalo en el editor **MakeCode**. ¿Hizo lo que esperabas? Si no se oprime el botón, ¿funciona?
3. Ahora prográmalo para que cuando presiones el botón **A** haga lo que acaba de hacer y que cuando presiones **B**, la flecha gire en dirección contraria. Para realizar este programa usa el mismo bloque **para siempre**.
4. ¿Qué pasa cuando presionas A y muy rápidamente B? ¿Es lo que esperabas?



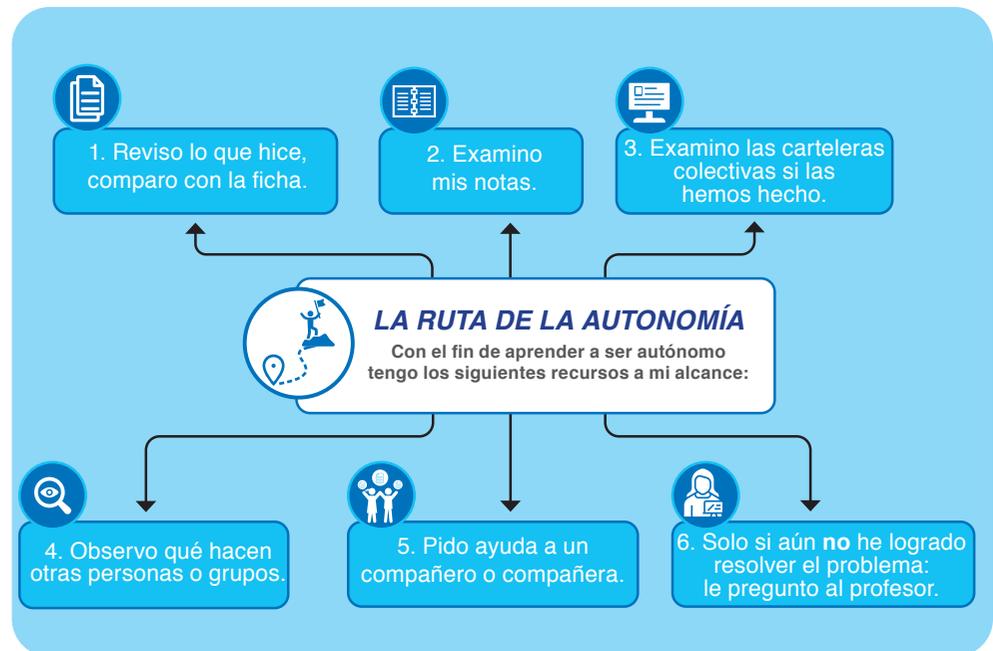


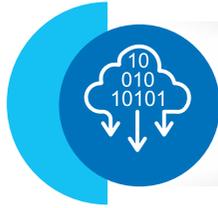
**Si tienes una *micro:bit* a tu alcance es el momento de probar tu programa. Sigue las instrucciones a continuación:**



Ya has trabajado con el simulador. Si tienes la oportunidad de tener una **micro:bit** podrás transferir el programa y probarlo igualmente en la tarjeta. Para ello:

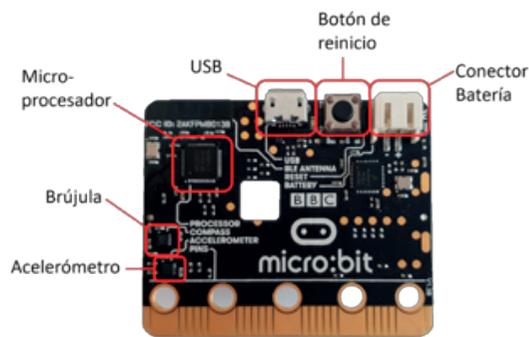
1. Conecta la **micro:bit** utilizando el cable USB a un puerto USB de tu computador. La **micro:bit** se encenderá y un disco externo aparecerá en tu computador.
2. La caja de pilas solo la necesitarás cuando quieras que la **micro:bit** funcione sin conexión al computador.
3. Utiliza el botón **Descargar** de la parte inferior del editor, guarda el archivo y luego coloca este archivo en la **micro:bit** que aparece con un disco llamado **MICROBIT**.
3. Podrás desconectar la **micro:bit** sin perder el programa que hayas cargado en ella. Si conectas la batería, el programa funcionará.
4. Es tu turno, ¿funciona? La **micro:bit** puede ahora funcionar autónomamente sin conexión al computador.





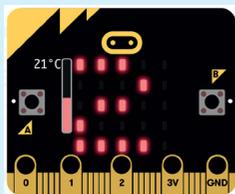
### Trabajando con otras entradas

1. 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. Esta es una **variable de entrada** también. Es una **variable numérica**, la cual asume varios valores.



```

para siempre
  mostrar número temperatura (°C)
  
```



```

para siempre
  mostrar número temperatura (°C)
  mientras temperatura (°C) > 25
    ejecutar
      mostrar número temperatura (°C)
      mostrar ícono
  
```

2. Antes de entrar al problema de aplicación, te sugerimos implementar este pequeño programa que se observa a la izquierda, primer código.
3. ¿Qué esperas que pase cuando lo ejecutes? Verifica en el simulador de la **micro:bit**; se trata de un **bloque** que repite lo que está dentro para siempre. Igualmente estás utilizando una nueva entrada, la **temperatura**. Este es un nuevo tipo de variable que no es **booleana** (definida en la ficha 1), sino una **variable numérica** que contiene la temperatura que mide el procesador de la **micro:bit**.
4. ¿Te muestra la temperatura en el arreglo de LED? Puedes variar la temperatura en el simulador cambiando la altura de la barra del termómetro. A medida que lo haces cambiará el valor de la temperatura, como se indica en la parte superior izquierda del termómetro.
5. Ahora complementa lo que le falta al programa para tener el código de la izquierda. Nuevamente, predice cuál será el resultado de su ejecución y verifica lo que pasará. Si quieres modificar la temperatura para probarlo puedes variar el valor en el termómetro.
6. ¿Se puede leer sin problemas el número? Si no logras leerlo bien, ¿qué se puede hacer? Estás **depurando** el programa.
7. Complementa este programa haciendo que salga una cara triste a una temperatura menor de 23 grados, simula y prueba.
8. A menos de 23 grados deberías ver la cara triste; entre 23 y 25 solo la temperatura; y a más de 25 grados la cara feliz y la temperatura.





### Aplicando lo aprendido



Muchas especies de tortugas marinas están en peligro de extinción. Por eso, los biólogos de la conservación y muchos voluntarios 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 se desarrollen los embriones.

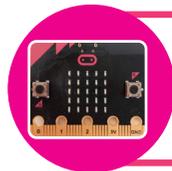


Un centro de preservación de tortugas marinas te pide ayuda programando un dispositivo, que les permita mantenerse informados sobre la temperatura del sitio de incubación; quieren saber si es muy baja, adecuada o muy alta para el desarrollo de los huevos.

Tu misión será programar la **micro:bit** para que detecte la temperatura e informe el valor en el tablero de LED. Además, deberá avisar a los biólogos 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 quieres, puedes inventar iconos que reemplacen los textos pero que resulten evidentes para quien observa.

Para empezar, puedes intentar hacer un diagrama de flujo sobre lo que debe hacer el programa. Luego usa el **MakeCode** para escribir estas instrucciones en el lenguaje de bloques.

Usa el simulador para ver si tu programa funciona apropiadamente.



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**.

### INFORMACIÓN COMPLEMENTARIA

Un procesador tiene variables de entrada. Por ejemplo:

- Un botón o pulsador es una variable **booleana** ya que puede estar oprimido o no: si está oprimido su valor es verdadero, si no está oprimido su valor es falso.
- Un sensor permite medir **variables** físicas como la **temperatura**, que pueden tomar muchos valores.
- Un **procesador** también puede tener salidas **booleanas**.
- Un LED, que también es **booleano**, puede estar encendido (verdadero) o apagado (falso).



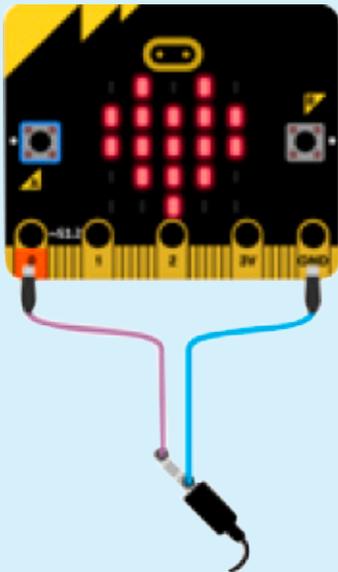


### Para ir más lejos

Los sistemas de alarma no solo informan a través de mensajes en una pantalla, sino que anuncian usando algún sonido. ¿Puedes mejorar tu programa de modo que haya un sonido cuando la temperatura sea superior al máximo y otro sonido cuando baje del mínimo admisible?

**Ayuda:** usa un comando de sonido como el que se muestra abajo. Debes conectar un parlante pequeño o audífono a tu **micro:bit**, como se muestra a la izquierda.

reproducir tono Do medio por 1 pulso



### Lo que hemos aprendido

Revisa y completa la siguiente tabla marcando una X en la columna que mejor represente tu aprendizaje:

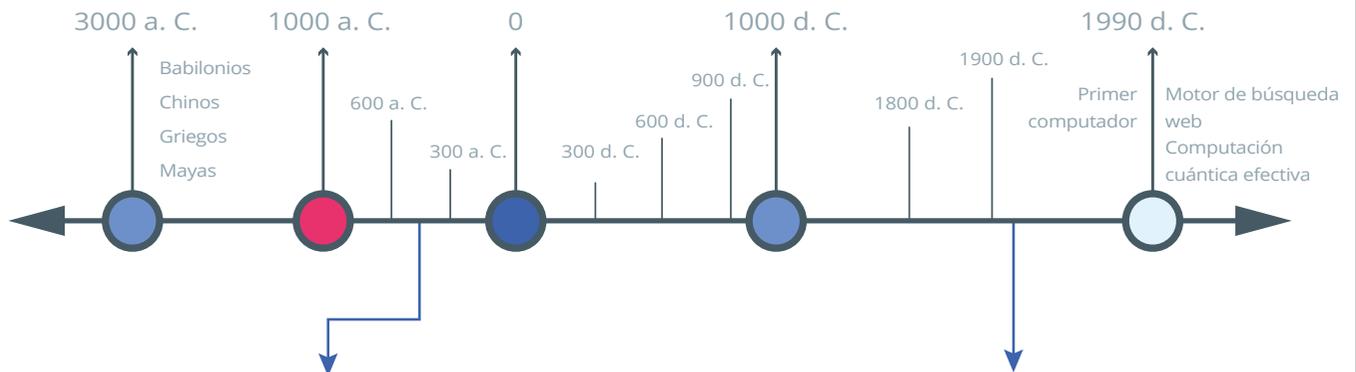
Verifica los aprendizajes logrados	Sí	Algo	No
Utilizo <b>condicionales</b> para decidir realizar o no una acción.			
Utilizo <b>condicionales</b> para controlar la repetición de un conjunto de acciones.			
Interpreto y hago <b>diagramas de flujo</b> sencillos.			
Utilizo <b>variables</b> de <b>entrada</b> de magnitudes físicas como la <b>temperatura</b> .			
Muestro una variable numérica, como la temperatura, en el arreglo de LED.			
Cargo un programa en la <b>micro:bit</b> y verifico su funcionamiento.			

Selecciona la opción que mejor represente tu opinión:

Contesta las siguientes preguntas	Sí	Algo	No
Las actividades realizadas fueron difíciles.			
Las actividades me motivaron.			
Siento que aprendí muchas cosas.			
Aún me quedan muchas dudas sobre lo que hice.			



### Un poco de historia



#### Pitágoras (569 a. C. - 475 a. C.)

Filósofo y matemático griego. Se considera como el primer matemático puro. Hizo aportes muy importantes a la Aritmética, Geometría, Música y Astronomía.

Estuvo fascinado por las proporciones. Aplicó su desarrollo de los números naturales para explicar la armonía de la lira, instrumento de cuerda de su época. Gracias a este estudio, cimentó las bases de toda la música occidental. Las armonías descritas por Pitágoras a menudo son detalladas mediante algoritmos y utilizadas como estrategia didáctica para transmitir el pensamiento computacional.

Su teorema más conocido es el que relaciona los lados de un triángulo rectángulo.



#### Margaret Hamilton (1936 d. C.)

Matemática, científica computacional e ingeniera de software estadounidense. En 1963, mientras trabajaba en el laboratorio Draper del MIT (Massachusetts Institute of Technology), lideró la División de Ingeniería de Software (término que ella misma acuñó).

Esta división tuvo a cargo el desarrollo del software de navegación para el computador ACG del programa espacial Apolo, que posteriormente permitiría que una nave tripulada llegara a la Luna. El ACG, que pesaba 32 kg, era el primero en utilizar circuitos integrados (chip). El programa diseñado por Hamilton contenía algoritmos de priorización de tareas, lo que permitió al módulo lunar alunizar a pesar de la saturación del computador y así jugó un papel crucial en el éxito de la misión. En 2017, Lego homenajeó a las mujeres de la NASA produciendo figurillas de juguete, entre otras, la de Margaret Hamilton.

Las bases del pensamiento computacional vienen desde la antigüedad y las mujeres han tenido un papel importante



### Mujeres y hombres que se destacan en la computación en Colombia y en el mundo



## Yury Niño

Yury Niño Roa estudió en el Instituto Técnico Industrial Nacional en Garagoa Boyacá donde pasó los primeros 15 años de su vida, luego estudió su pregrado y maestría en Ingeniería de Sistemas y Computación en la Universidad Nacional de Colombia.

Ella es una de las líderes mundiales en Ingeniería del Caos, una disciplina que busca mejorar la confiabilidad de los sistemas computacionales simulando la ocurrencia de fallas.









Anexo

3

Juego de lógica y condiciones



Objetivo del juego

Como vimos en esta ficha, utilizamos condiciones en la programación como estructuras que controlan qué parte del código se ejecuta y cuál no, según la situación.

Este juego puedes jugarlo con 1 a 3 personas más y toma entre 15 y 25 minutos.



Instrucciones

En las siguientes páginas se encuentra el tablero (2 hojas a unir y colocar sobre un cartón) y las tarjetas A, B y de sorpresa.

- Se juega con un solo dado.
- Se deben barajar y organizar en el lugar indicado en el tablero las tarjetas A, B y Sorpresa, colocadas boca abajo.
- Cada persona en el juego tiene una ficha que colocará en la casilla de inicio.
- Gana quien llegue primero a la casilla final.
- Para comenzar todas las personas lanzan el dado y quien obtenga el número más pequeño comienza. El segundo será el segundo valor más pequeño y así para el resto si juegan más de 2.
- Quien tiene el turno lanza el dado y avanza su ficha ese número de casillas.
- Deberá interpretar la condición en la que cae utilizando las tarjetas A, B o Sorpresa y realizar la acción indicada. Para esto deberá tomar la tarjeta que se encuentra encima del mazo. Una vez utilizada la tarjeta se coloca debajo de todas las tarjetas.
- Si la ficha llega a una casilla realizando una acción diferente a avanzar o retroceder el número que indica el dado, no debe volver a evaluar la nueva condición, sino esperar a su turno para lanzar el dado y avanzar.



Variante del juego

- En lugar de usar un dado, crea un dado con la micro:bit que al sacudirla arroje un número entre 1 y 6.

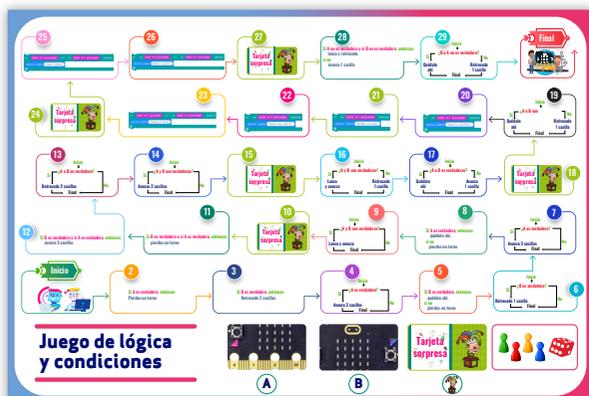
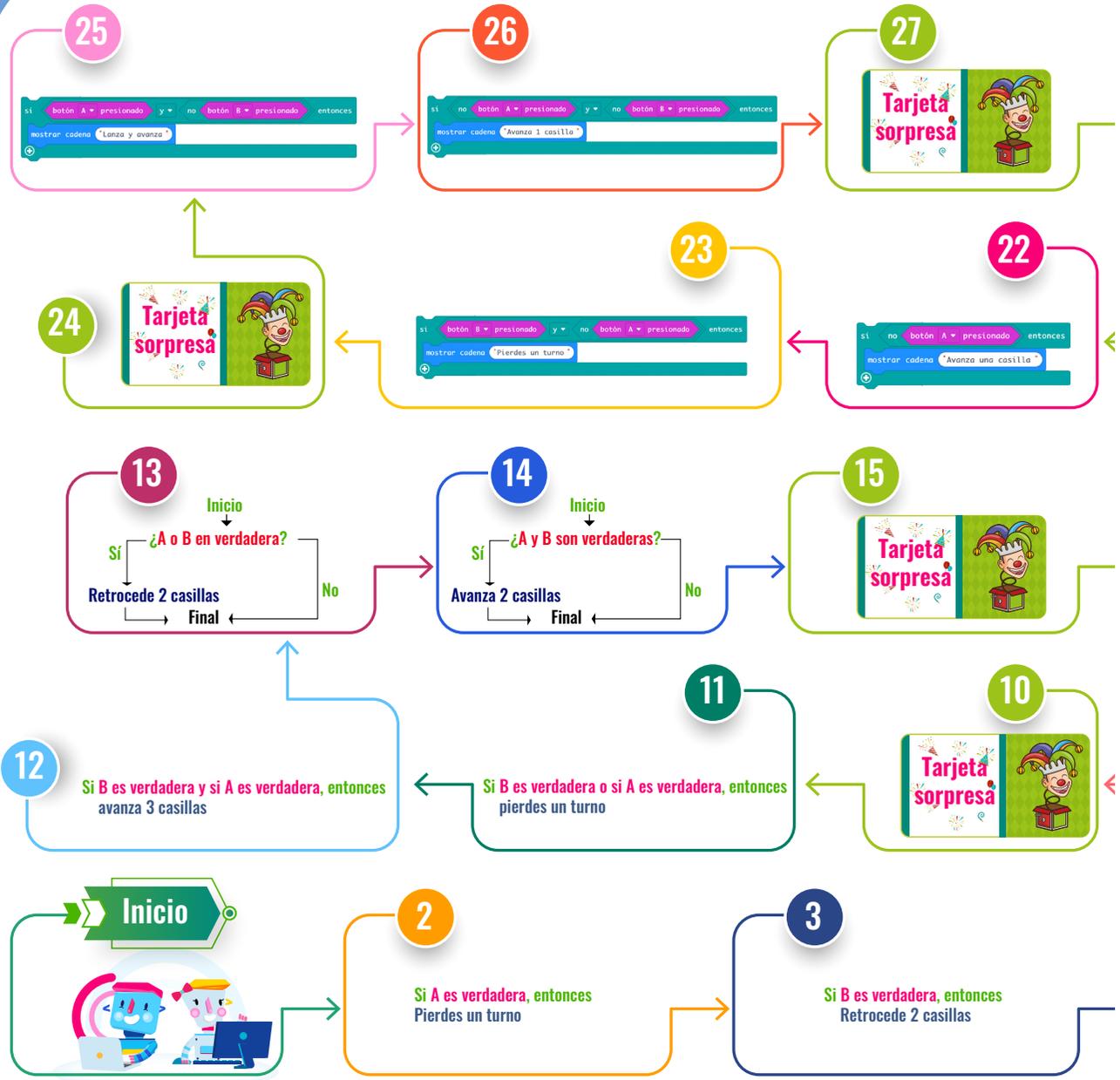
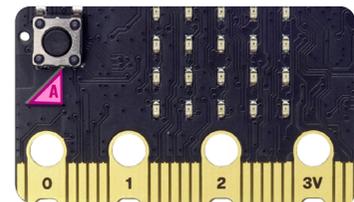


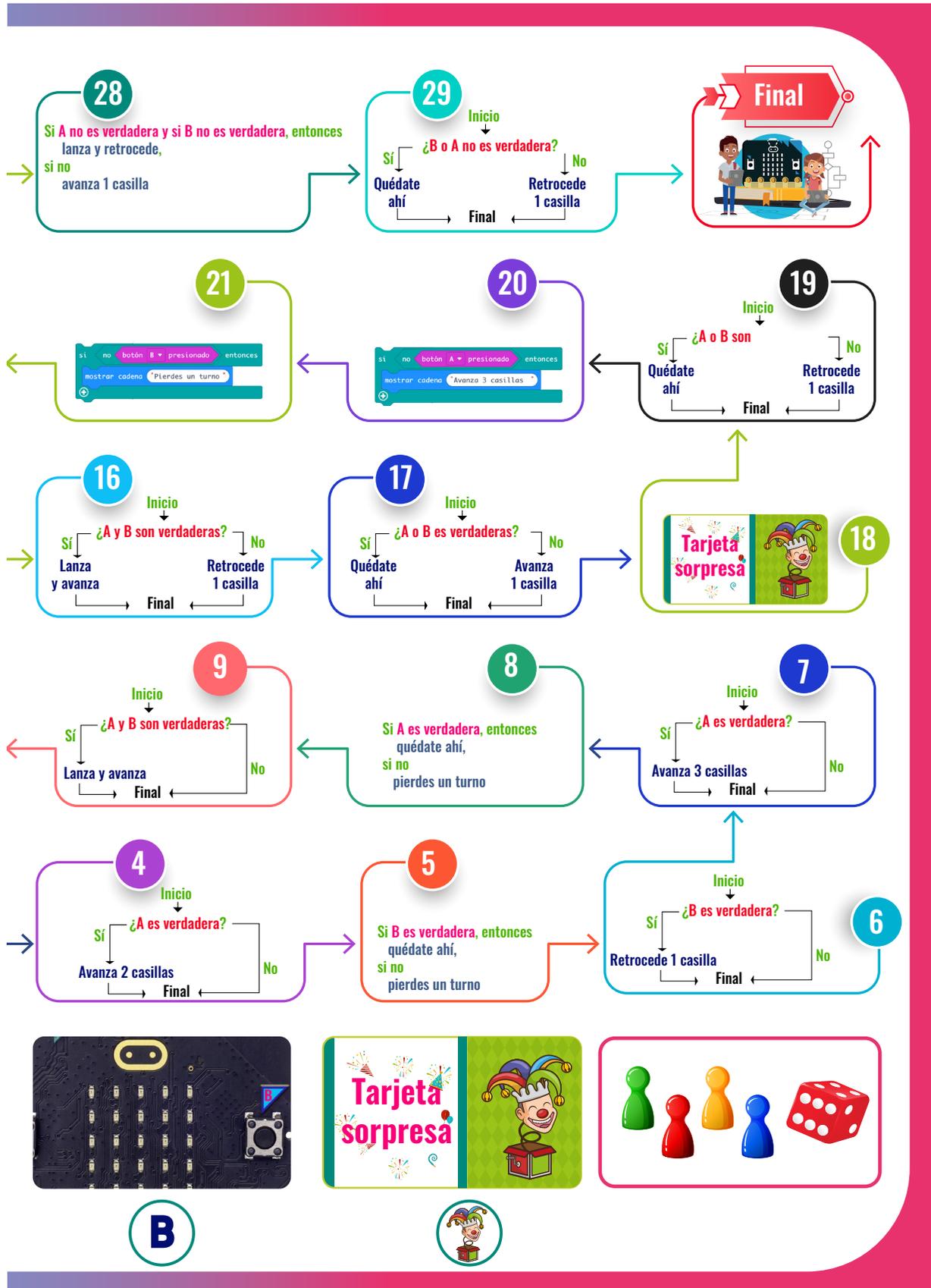
Tabla con ejemplo de valoración de condiciones con "y", "o" y "no"

A	B	A y B	A o B	A y no(B)
V	V	V	V	F
V	F	F	V	V
F	V	F	V	F
F	F	F	F	F



# Juego de lógica y condiciones







### Tarjetas sorpresa

Si estás de primero en el juego entonces te quedas en tu lugar, si no, avanzas hasta el lugar del jugador más avanzado.



Si tu edad es mayor que la de los demás jugadores entonces debes saltar en un pie tantas veces como tu edad.



Si usas gafas o usas lentes de contacto, entonces los demás jugadores te deben dar un detalle.



Si estás de primero en el juego y está lloviendo en este momento, entonces retrocede hasta el lugar del jugador más atrasado.



Si tienes ojos café y tienes pelo crespo, entonces debes cantar una canción divertida al resto de jugadores.



Si tu edad es menor de 12 años, entonces debes contar un chiste al resto de jugadores.



Si es por la mañana y tu edad no es mayor de 12 años, entonces puedes ayudar a avanzar 2 casillas a cualquier otro jugador.



Si estás de último en el juego, avanza 3 casillas.



Si te gusta programar, puedes sacar otra carta de sorpresa, si no, debes hacer 10 abdominales.



Si tu estatura es mayor que la de los demás jugadores puedes dar una tarjeta sorpresa a otro jugador.



Si estás de último en el juego, entonces puedes lanzar 2 veces el dado y avanzar la suma de los dos lanzamientos.



Si estás usando suéter o estas usando chaqueta, entonces debes usarlo al revés hasta el final del juego.



Si ya has tomado una carta de sorpresa antes, entonces debes imitar a un animal de tu preferencial.



Si acabaste de sacar un número par en el dado, entonces debes cantar una canción que tenga la palabra "vida", si no logras cantarla, pierdes un turno.



Si está lloviendo y te gusta programar, entonces los demás jugadores deben hacer 5 sentadillas cada uno.



Si puedes decir cuál es la capital de Uruguay, entonces te quedas en tu lugar, si no debes retroceder a tu última posición en el tablero.



Si usas gafas y usas zapatos deportivos, entonces debes dar un detalle a los demás jugadores



Si sacaste un número impar en el dado, debes retroceder esa cantidad.



Si te gusta programar y estás aprendiendo muchas cosas al respecto, entonces date un aplauso y avanza 3 casillas más.



Si puedes decir cuál es la capital del departamento del Atlántico, entonces puedes pedir a otro jugador que cuente un chiste.





# Tarjetas A

Está lloviendo en este momento.	Está haciendo sol en este momento.	Hoy es tu cumpleaños.	Usas gafas.
Llevas puesta alguna prenda roja.	Tu edad es un número par.	Tienes ojos de color café.	Naciste en el mes de febrero.
Eres zurda/o.	Llevas puesta alguna prenda azul.	Vives en un lugar cerca del mar.	Hoy es jueves.
Tu escuela tiene campaña de separación de residuos en la fuente.	Tu edad es un número impar.	Eres diestra/o.	Naciste en el mes de abril.
Hoy es martes.	Estás usando medias.	Ahora el día está nublado.	Tienes solo un nombre.
Naciste en un año par.	Naciste en un año impar.	Sabes un chiste de gatos.	Sabes una canción con la palabra "camino".
Puedes recitar los números de 0 a 9 en inglés.	Puedes traducir al español "coding for kids".	Puedes decir los 4 primeros números primos.	Sabes qué significa la sigla STEM.





### Tarjetas B

Estudiantes en el salón > 30	Tu edad < 14 años	Número hermanos = 2	Tu estatura > 140 cm
Tu estatura < 140 cm	Altura sobre el nivel del mar de la escuela < 1000 metros	Día del año > 200	Semana del año < 20
Suma edad jugadores > 20	Suma edad jugadores < 30	Día del año < 130	Semana del año > 40
Promedio edad jugadores < 14	Estudiantes en el salón < 20	Talla de zapatos > 34	Doble de tu edad > 25
Talla de zapatos < 32	Día de la semana > 3	Mes del año < 5	Mes del año > 7
Hora del día > 10:00	Número hermanos < 2	Día de la semana > 3	Altura sobre el nivel del mar de la escuela > 1200 metros

