2 SALVANDO A LAS TORTUGAS

Aprendizajes

Al final de esta actividad se espera que puedas:

- Utilizar bucles para ordenar la repetición de un conjunto de acciones.
- Reconocer que un bucle se puede repetir indefinidamente, un número de veces o mientras se cumpla o no se cumpla una condición.
- Interpretar y hacer diagramas de flujo sencillos.
- Cargar el código en la micro:bit y verificar el funcionamiento del programa.
- Utilizar variables continuas de entrada como la temperatura.

SESIÓN 1

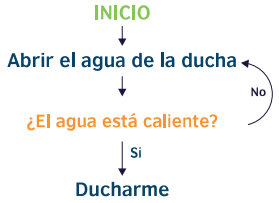
QUÉ SABEMOS, QUÉ NECESITAMOS SABER

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 para programar una **micro:bit** en el editor **Makecode**.

Existen diferentes códigos o lenguajes que sirven para programar diferentes **procesadores**, pero todos describen un proceso lógico de pasos o instrucciones. Algunas acciones requieren pocos pasos, como mostrar el corazón en el tablero de **LEDs** de la **micro:bit,** mientras otros 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.

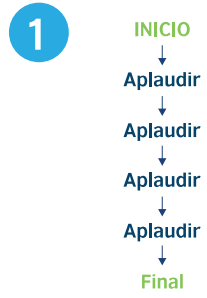
Los programadores usan **algoritmos** expresados en **diagramas de flujo** para representar procesos que en muchas instrucciones. Así se pueden organizar las instrucciones en el orden correcto para luego realizar correctamente la **programación.**

Un proceso cualquiera, como por ejemplo bañarse, puede expresarse en un **diagrama de flujo** como el de la derecha. Normalmente se inicia con alguna acción (abrir la llave por ejemplo) y usualmente hay un momento en que se toma una decisión en función de la temperatura del agua; por ejemplo, si el agua está caliente me ducho, pero si no, espero un rato y vuelvo a revisar si el agua está caliente, de modo que se hace una especie de ciclo o "bucle" en el proceso.

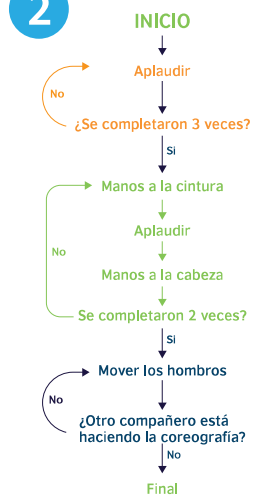


Piensa en un proceso sencillo que hagas en tu vida cotidiana; por ejemplo, preparar un huevo frito o tender la cama. ¿Puedes expresar este proceso como un diagrama de flujo? Dibuja el esquema y compártelo con tus compañeros. ¿Qué piensan ellos de tu diagrama?

1



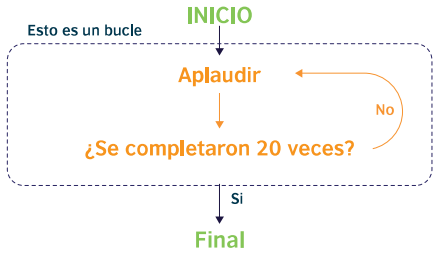
2



DESCONECTADOS

Tú y tu equipo deberán trabajar para reproducir las instrucciones de un diagrama de flujo. Reúne a tu grupo y asignen los roles que van a desempeñar.

1. Ejecuta el algoritmo que se encuentra a la izquierda.
2. Si tuviésemos que aplaudir muchas veces, por ejemplo 20, podríamos colocar 20 veces la instrucción **Aplaudir** o simplemente:



3. Ahora sigue el algoritmo representado en el diagrama de flujo 2. ¿Funciona? ¿Funcionó para todos los grupos? Si algo parece incorrecto, tienes razón. Se requiere del depurador para corregir el error. ¿Qué se podría hacer? Discútelo con tus compañeros y con el docente.
4. Habrás encontrado que algunos **bucles** se repiten **3 veces** o **2 veces,** pero también aparece un bucle diferente que se repetirá mientras esta condición sea cierta: **¿Otro compañero está haciendo la coreografía?**
5. Este bucle se repetirá mientras la condición sea cierta y como en la clase tienes muchos compañeros, probablemente te tengas que quedar moviendo los hombros indefinidamente; dado que tus compañeros también te verán mover los hombros y también deberán continuar moviéndolos.
6. ¿Ya encontraste una solución a este problema? Si la encuentras hiciste la **depuración** del algoritmo.

RESUMEN
Has visto dos tipos de **bucles**, o lazos, o repeticiones:
El **bucle** que repite un conjunto de instrucciones por un **número de veces.**
El **bucle** que repite un conjunto de instrucciones **mientras una condición sea verdadera**, esto es, se cumpla.

7. Ahora es tu turno de hacer tu primer diagrama de flujo a continuación.

Traduciendo a la micro:bit

Todo programa en la micro:bit debe estar en un bucle de repetir general como el siguiente (hay otros):

Igualmente en la micro:bit tenemos bucles que se repiten un número de veces:
Finalmente en la micro:bit también se tienen bucles que se repiten mientras una condición sea cierta:

En este caso, se vería una flecha apuntando al norte mientras el botón A esté presionado.
Ya te habrás dado cuenta que las condiciones que son ciertas o falsas tienen esta forma

APLICANDO LO APRENDIDO

Un **robot que cocina** que tienes comprende un **programa** en un **diagrama de flujo.** A continuación se presenta una receta para hacer una deliciosa torta de banano. Deberás hacer el **algoritmo** representado en un **diagrama de flujo**, con lo que ya sabes, para **programar** el **robot** de cocina. Utiliza **bucles**.
Una vez termines, busca un compañero o el docente para que haga la labor del **depurador**. No queremos que el robot haga locuras

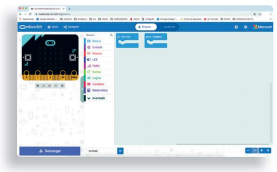
Ingredientes:
- 200 g o 1 ¾ tazas de harina
- 2 cucharaditas de levadura química
- ¼ de cucharadita de bicarbonato
- ½ cucharadita de sal
- 75 g o 1/3 de taza de margarina
- 140 g o 2/3 de taza de azúcar
- 2 huevos batidos
- 1 taza de bananos maduros (2 o 3)

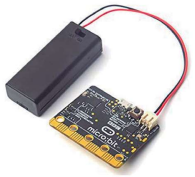
Preparación (hacer el diagrama de flujo):
Aplastar los bananos con un tenedor mientras no sean puré.
Prender el horno y esperar hasta que esté a 350°F.
Poner en un recipiente la levadura química, la harina, la sal y el bicarbonato.
Revolver mientras no esté todo mezclado.
Agregar la margarina derretida.
Revolver mientras no esté todo mezclado.
Agregar lentamente el azúcar revolviendo mientras haya azúcar por añadir.
Agregar los huevos batidos revolviendo.
Agregar los bananos revolviendo mientras no esté bien mezclado.
Engrasar un molde y verter la masa adentro.
Cocinar por espacio de 1 hora y 10 minutos aproximadamente.

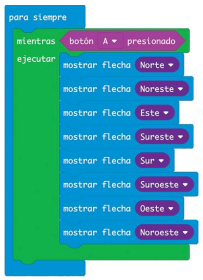
RESUMEN
Según el objetivo que tengas, puedes utilizar diferentes tipos de **bucles:**
• Los Bucles indefinidos o para siempre que repiten el conjunto de instrucciones indefinidamente, o hasta que se detenga el **procesador.**
• **Bucles** que se repiten un número exacto de veces.
• **Bucles** que se repiten mientras una condición sea cierta, por ejemplo mientras el botón B esté oprimido.
La instrucción **mostrar cadena** mostrará un texto. La instrucción **mostrar flecha** mostrará una flecha en la dirección que se indica.
La condición **botón A presionado** será **verdadera** si A está presionado.

El futuro digital es de todos MinTIC

<PROGRAMACIÓN/ PARA NIÑOS Y NIÑAS







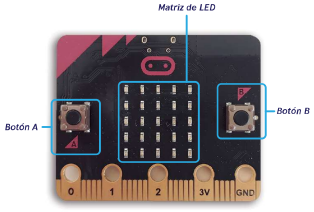
SESIÓN 2

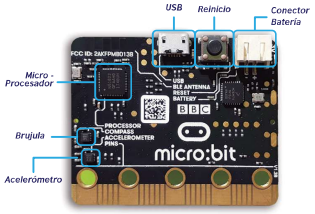
CONECTADOS: MANOS A LA MICRO:BIT

Conectar la micro:bit al computador y al editor.

Es el momento de conocer la **micro:bit.** Ya has explorado el entorno de programación y viste cómo actúa el dispositivo usando el simulador. En esta sesión podrás usar el **procesador** real de la **micro:bit.**
Para trabajar con la Micro:bit necesitarás:

1. Entra a **Makecode** en tu computador o al **editor** en línea si tienes internet.
2. Conecta la **micro:bit** utilizando el cable USB (cable de la derecha) a un puerto USB de tu computador. La **micro:bit** se encenderá y un disco externo aparecerá en tu computador.
3. Ya puedes programar la micro:bit. La caja de pilas sólo la necesitarás cuando quieras que la micro:bit funcione sin conexión al computador. Podrás desconectar la micro:bit sin perder el programa que hayas cargado en ella.
4. Identifica en la micro:bit los siguientes elementos:
   1. Botón A
   2. Botón B
   3. El arreglo de 25 LEDs (5x5) de la **micro:bit**
5. Por el reverso del micro:bit identifica:
   1. El procesador
   2. La brújula (compass)
   3. El acelerómetro





Es hora de programar, simular y cargar el programa en la micro:bit y ver si este programa funciona.
1. Este **programa** que se encuentra a la izquierda. ¿Qué crees que hace?
2. Ahora prográmalo en el editor del **Make-code.** ¿Hizo lo que esperabas? Si no se oprime el botón ¿funciona?
3. Utiliza el botón Descargar de en la parte baja del editor, guarda el archivo y luego coloca este archivo en la micro:bit que aparece con un disco llamado MICROBIT. ¿Funciona?

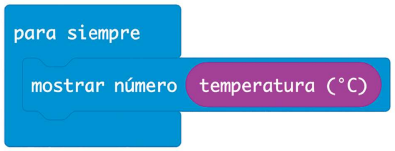
Generar autonomía

Con el fin de aprender a ser autónomo tengo los siguientes recursos a mi alcance:

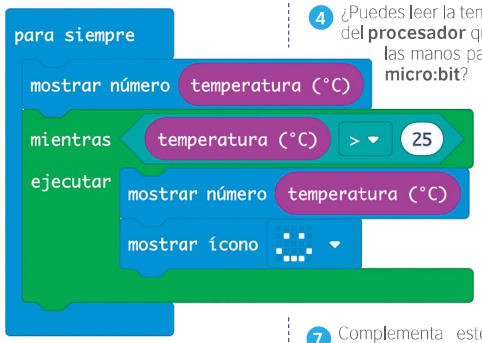
1. Reviso lo que hice, comparo contra la ficha
2. Examino mis notas
3. Examino las carteleras
4. Observo que hacen mis compañeros
5. Le pregunto a un compañero, y si aún no he logrado resolver el problema
6. Le pregunto al profesor

Hacer un programa, simularlo, cargarlo y probarlo

1. Es el momento de modificar el programa. El objetivo ahora es que la flecha realice 5 vueltas en un sentido de las manecillas del reloj y 3 vueltas en el otro sentido, y que esta secuencia la repita para siempre. ¿Funcionó? Recuerda que si no funciona tienes varios recursos (ver tabla de la izquierda).
2. Antes de entrar al problema de aplicación te sugerimos implementar este programa.



3. ¿Qué esperas que pase cuando lo ejecutes? Verifica en el simulador y sobre la **micro:bit**; se trata de un **bucle** que se repite siempre. Igualmente estás utilizando una nueva entrada, la **temperatura.** Este es un nuevo tipo de variable que no es **booleana**, sino una **variable numérica** que contiene la temperatura que mide el procesador de la micro:bit.
4. ¿Puedes leer la temperatura? No es exactamente la temperatura del salón, sino la del **procesador** que está ligeramente por encima. Si encierras la **micro:bit** entre las manos para calentarla, ¿qué ocurre con la temperatura que muestra la **micro:bit**?



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 calentar la **micro:bit** colocándolo entre las manos, o enfriándolo, dejándola sobre una mesa sin tocar y soplando con un cartón.
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, carga y prueba.
8. Con menos de 23 grados deberías ver la cara triste, entre 23 y 25, solo la temperatura y más de 25 grados, la cara feliz además de la temperatura.

Un poco de historia

En 1937, Howard Aiken, un estudiante de la universidad Harvard (EE.UU.) se imaginó un calculador para resolver sus ecuaciones. Cuando presentó su idea, un técnico de la universidad recordó que en una bodega había una máquina parecida, olvidada por todos: el calculador analítico de Charles Babbage. Su hijo Henry lo había construido parcialmente y regalado a la universidad en 1875.

Aiken hizo construir un nuevo calculador por la recién creada empresa llamada IBM, de acuerdo a los planos de Babbage, pero con algunas mejoras. Se lo entregó a la universidad de Harvard en 1944 donde la llamaron Harvard Mark I. Fue uno de los dos primeros computadores construidos en el mundo.

Los primeros programas fueron desarrollados por un pequeño grupo de 3 personas incluyendo una joven profesora de matemática, Grace Hopper, que en 1951 concibió el primer compilador (traductor a lenguaje de la máquina) y el lenguaje Cobol en 1959. En la actualidad, hay un importante evento anual en su nombre, donde se encuentran miles de mujeres que siguen contribuyendo con grandes avances en la computación:

https://ghc.anitab.org/calendar/2019-grace-hopper-celebration/

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 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 LEDs. 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.

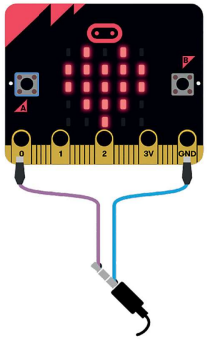
Para empezar, pueden intentar hacer un diagrama de flujo sobre lo que debe hacer el programa. Luego usa el **Makecode** para escribir estas instrucciones en el lenguaje de bloques.

Usa el simulador para ver si tu programa funciona apropiadamente y descárgalo a 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.**

RESUMEN
Un procesador tiene variables de entrada. Por ejemplo:
- Un botón es booleano ya que puede estar oprimido o no: si está oprimido su valor es verdadero, si no está oprimido su valor es falso.
- Un sensor que suministra la temperatura, es una variable que es continua, un número decimal.

Un procesador también puede tener salidas booleanas:
- Un LED, que también es booleano, puede estar encendido (verdadero) o apagado (falso).

Fuente imagen: Grace Hopper IMAGEN DE LYNN GILBERT, CC BY-SA 4.0. hsps://encrypted-tbn0.gsta(c.com/images?q=tbn:ANd9GcT0SkdxGvSWJrLI6jRsNJlmW2PckO_AcI7W-DnUgV8SQ63DQRgc



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.

QUÉ HEMOS APRENDIDO

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

VERIFICA LOS APRENDIZAJES LOGRADOS	SI	ALGO	NO
Utilizar **bucles** para ordenar la repetición de un conjunto de acciones.			
Reconocer que un bucle se puede **repetir indefinidamente, un número de veces** o mientras se cumpla o no se cumpla una condición.			
Interpretar y hacer d**iagramas de flujo** sencillos.			
Cargar el código en la **Micro:bit** y verificar el funcionamiento del programa.			
Mostrar una variable continua como la **temperatura en el arreglo de LEDs.**			

Selecciona la opción que mejor representa su opinión:

CONTESTA LAS SIGUIENTES PREGUNTAS	SI	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.			

3 EXPEDICIÓN ESPELEOLÓGICA

Aprendizajes

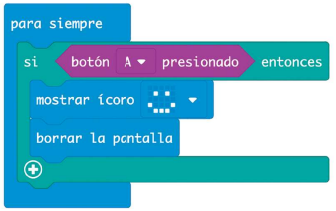
Al final de esta actividad se espera que puedas:

- Utilizar variables booleanas de entrada que simulan la acción de sensores.
- Comunicar instrucciones utilizando la pantalla de LEDs y un código de flechas.
- Interpretar una secuencia de instrucciones para resolver un problema como el de un laberinto.
- Interpretar un diagrama de flujo para resolver problemas como el de un laberinto.
- Utilizar operaciones lógicas para decidir qué acción se ejecuta.
- Utilizar lazos que se repiten hasta terminar la tarea.

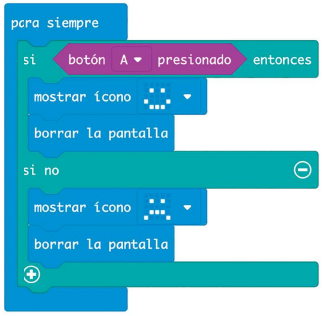
SESIÓN 1

QUÉ SABEMOS, QUÉ NECESITAMOS SABER

En las fichas anteriores ya hemos trabajado **entradas booleanas** (los botones) y la salida de LEDs. Igualmente, hemos utilizado bloques que representan algunas acciones o instrucciones que se deben repetir.
Las **variables booleanas** pueden asumir dos valores solamente: **verdadero** o **falso.**
Cuando el **Botón A** está oprimido, su valor es **verdadero** y cuando no, es **falso.** En este ejemplo, si presionamos A veremos una cara feliz.



Esta es una nueva forma de controlar la realización o no de ciertas instrucciones. En este caso se ejecutará la instrucción **mostrar cara feliz** si **A está presionado.** ¿Con qué objetivo se coloca borrar pantalla? ¿Qué sucede si no se coloca esta instrucción?



Ahora mira el diagrama y programa en bloques de la izquierda:

En este caso, al no estar oprimido el botón A, se verá una **cara triste**, mientras que si lo oprimimos, verás una **cara feliz.** Esta estructura de hecho se puede complicar aun más si se oprime el símbolo más, quedando el bloque como se muestra a la izquierda.
Si se oprime el + con el ratón, la estructura permite anidar de nuevo otra condición.

¿Se requiere en este caso colocar borrar pantalla? ¿Por qué?