

Sesión

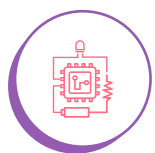
3

Aprendizajes esperados

Al final de esta sesión verifica que puedas:



Leer y escribir datos de dispositivos externos usando los pines.



Cambiar de escala de las lecturas a una escala apropiada a la naturaleza de la variable.

Duración sugerida



40%

40%

20%

Material para la clase

- Acceso a MakeCode.



Lo que sabemos, lo que debemos saber



Esta sección corresponde al 40% de avance de la sesión

Como probablemente sabes, todo circuito eléctrico requiere dos puntos de conexión para que funcione, con el fin de establecer un camino para la corriente eléctrica.

En las sesiones anteriores lograste leer datos de variables físicas como la temperatura y el nivel de sonido, y sabes cómo crear variables para guardar estos datos. También aprendiste a usar un contador para llevar un registro de cuántas mediciones has realizado.

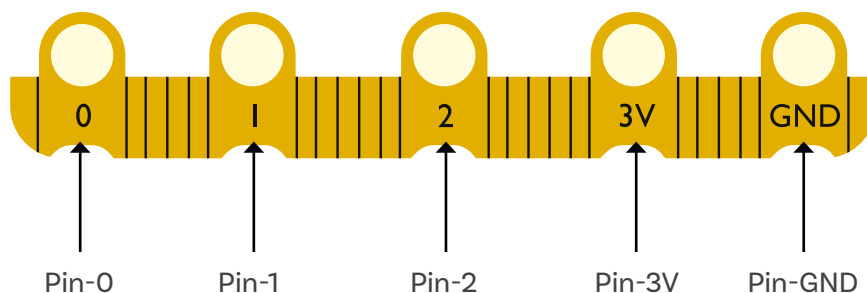
Además, descubriste cómo interpretar datos que la *micro:bit* obtiene en unidades para la temperatura o grados para la orientación, y cómo leer otras variables, como el nivel de sonido o de luz, en una escala del 0 al 255.

En esta sesión vas a explorar el uso los pines de la *micro:bit* y entender para qué sirven.

Igualmente, aprenderás a cambiar valores de 0 a 255 a una escala apropiada, por ejemplo, de 0 a 100.

La *micro:bit* cuenta con un conjunto de 25 pines o puntos de conexión eléctricos en la parte inferior, ver *Figura 1*.

Figura 1. Pines de la tarjeta *micro:bit*



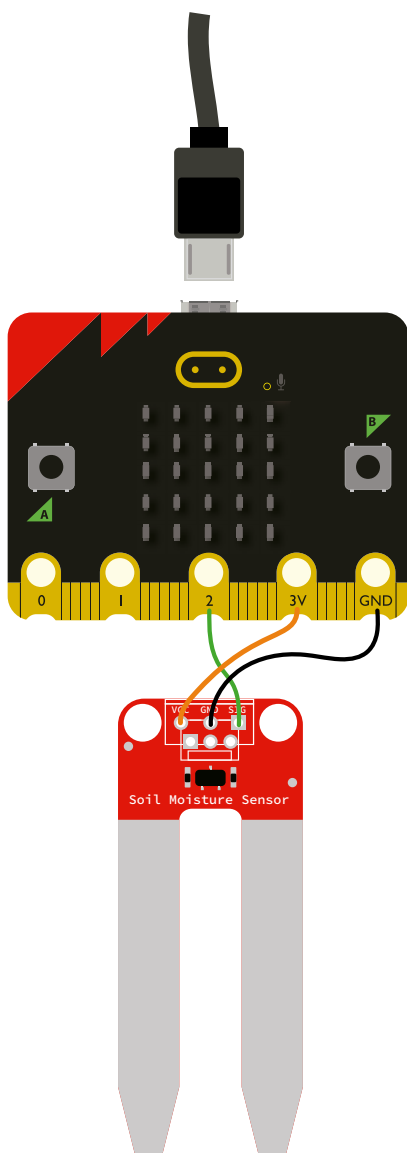
Hay 5 pines grandes que están conectados a los huecos de la tarjeta. Entre estos pines grandes, 3 que están numerados del 0 al 2, hay un pin etiquetado 3V (3 voltios) y hay un pin etiquetado GND, que permite conectar el circuito de retorno.



Nota

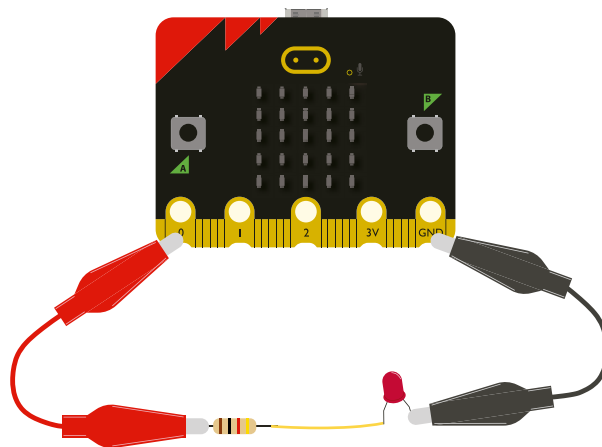
Mientras algunos sensores de la *micro:bit* trabajan con la escala 0 a 255, los pines trabajan con una escala que varía de 0 a 1023.

Figura 3. Conexión de un sensor de humedad a la *micro:bit*



Para conectar un elemento externo, como un nuevo sensor, este deberá conectarse entre un pin (0, 1 o 2) y el pin denominado GND. La Figura 2 muestra cómo se conectaría un **LED** a los pines.

Figura 2. Conexión de un LED a la *micro:bit*



En este caso, el pin-0 no se usa como entrada, sino como **salida**. El circuito eléctrico se completa al colocar en el pin-0 un valor de tres voltios, que es el máximo voltaje que emite la *micro:bit*. Esto se logra escribiendo sobre dicho pin el número 1023.

La pequeña **resistencia** que se coloca en este circuito limita la corriente eléctrica para que no se dañe el pequeño LED rojo que se muestra en la Figura 2. Finalmente, se conecta a GND para completar el circuito.

Si se fuera a conectar un sensor, por ejemplo, de humedad, la conexión sería como se muestra en la Figura 3. Una ampliación de la conexión de este sensor se muestra en la Figura 4:

Figura 4. Detalle de la conexión del sensor de humedad a la *micro:bit*

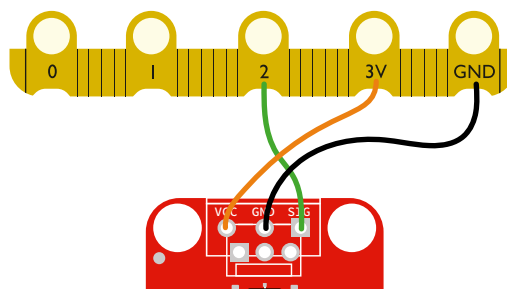
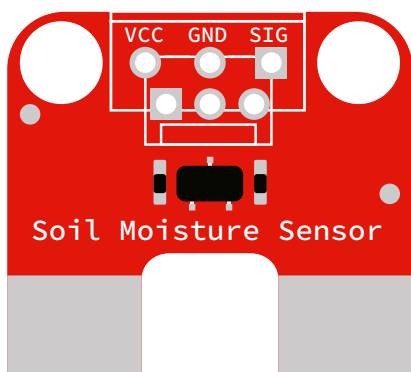


Figura 5. Puntos de conexión del sensor de humedad



Nota

Si no cuentas con un sensor de humedad como el que se muestra aquí, puedes improvisar uno usando un par de puntillas y unos cables cocodrilo. Podrás encontrar instrucciones detalladas para hacerlo en el dando clic sobre el código QR o escanéandolo:



El sensor requiere de tres puntos de conexión:

- **VCC:** donde se debe colocar un **voltaje** de 3 voltios para que funcione.
- **GND:** que es tierra y representa en punto de cierre del circuito eléctrico que se establece.
- **SIG:** que es la abreviación de la palabra en inglés “*signal*” o señal en español, que es donde se podrá “leer” el valor de humedad medido de un suelo de un cultivo, por ejemplo.

Con este ejemplo del sensor de humedad, veamos ahora como usar el PIN-2 para “leer” el valor de humedad detectado.

En la *Figura 6* se muestra dónde se encuentran en *MakeCode* las instrucciones para usar los pines para “leer” un dato.

Figura 6. Bloques del Menú de pines en MakeCode



Tendrás que oprimir en el panel de herramientas la opción **avanzado** para que aparezca el menú de pines y, al hacerlo, verás los bloques de instrucciones de uso de pines.

Es hora de trabajar con *MakeCode* y conocer el funcionamiento de algunas de estas instrucciones.

Glosario



Pin: punto de conexión eléctrica.



GND: es la abreviación de la palabra en inglés “ground”, que en español se traduce como tierra en un circuito eléctrico.



Voltio: uno de los términos usados para describir la electricidad.



LED: pequeño bombillo que se puede encender o apagar. La palabra LED viene de su nombre en inglés (*Light Emitting Diode*) que corresponde a diodo emisor de luz.



Salida: corresponde a un medio por el cual un sistema de computación comunica información, por ejemplo, la pantalla de LED o la emisión de un sonido en la *micro:bit*.



Resistencia: componente electrónico hecho de carbón u otros materiales que se opone al paso de la corriente eléctrica. Las franjas de colores de una resistencia indican su valor.



Voltaje: es la fuerza que hace que la electricidad se mueva en un circuito. Se mide en voltios (V).

Nota

Para colocar comentarios en un código, basta ubicar el cursor sobre el lugar deseado en el espacio de programación de *MakeCode* y oprimir el botón derecho del ratón. Al hacerlo, aparecerá la opción **añadir comentario**.

Manos a la obra**Conectadas**

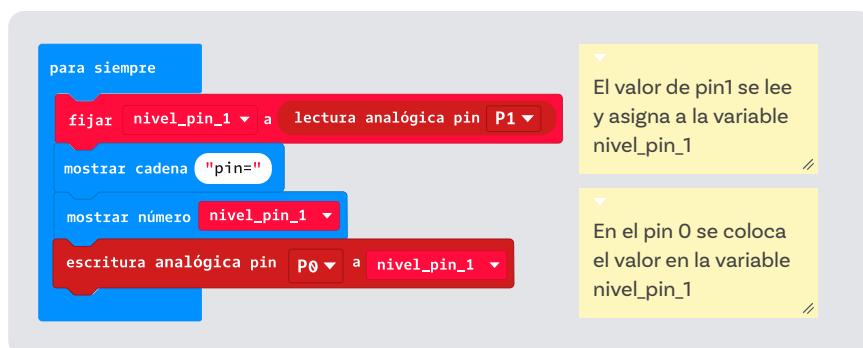
Esta sección corresponde al 80% de avance de la sesión

Por ahora no vas a trabajar con un sensor. Te ocuparás de hacer funcionar los pines como entradas, para leer valores en ellos, y de que trabajen como salidas, colocando valores que podrían permitir, por ejemplo, encender un LED, hacer funcionar un motor o hacer sonar una alarma.

Organícense en parejas, siguiendo las indicaciones de su docente. Accedan a *MakeCode*.

Empecen por examinar el programa de la *Figura 7*.

Figura 7. Programa de lectura y escritura analógica de pines



¿Qué hace?

Los comentarios les ayudarán a inferir lo que hace este programa. Colocar comentarios en un programa es importante para que puedan recordar lo que han hecho y también para que otras personas puedan entender más fácilmente sus programas.

Ahora introduzcan en *MakeCode* el programa de la *Figura 7* y verifiquen su funcionamiento.

Para cambiar el valor en el pin 1 pueden usar el cursor sobre ese pin. En la *Figura 8* se ha colocado el valor 258 en el pin 1, pero como observan aún no aparece el mismo valor en el pin 0. Sin embargo,

Figura 10. Bloques de uso de los pines: lectura y escritura digital y analógica



Como quizás notaron, al ejecutar el código de la Figura 7, mientras la *micro:bit* está mostrando un dato o unos caracteres en pantalla, el programa no continúa con la siguiente instrucción. Por eso toma un tiempo en salir en el pin 0 el valor almacenado en la variable Nivel_pin_1.



al esperar un momento, se observa algo parecido a lo que muestra en la Figura 9. Esto sucede porque el código asigna o “escribe” en el pin 0 el valor que había tomado o “leído” del pin 1.

En este caso, el pin 0 es un pin de salida, porque se coloca o escribe un valor en él, y el pin 1 es un pin de entrada, porque se lee un valor de él.

Figura 8. Pines 0 y 1 de la *micro:bit* mostrando valores diferentes

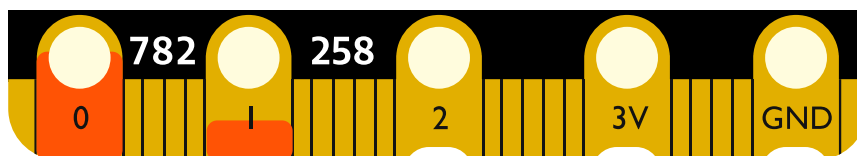
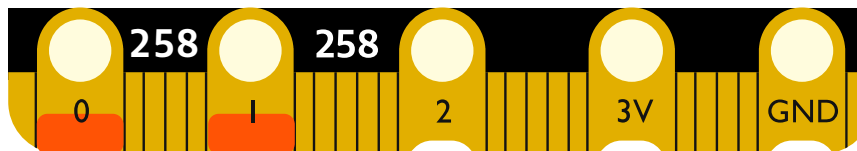


Figura 9. Pines 0 y 1 de la *micro:bit* mostrando el mismo valor



Los pines de la *micro:bit* son flexibles en su uso y tienen 4 formas de trabajar:

- Como entrada, leyendo un valor de 0 a 3 voltios que se convierte en un número de 0 a 1023.
- Como salida, escribiendo un número entre 0 y 1023 que se convierte en un voltaje de 0 a 3 voltios.
- Como entrada digital, que solo puede tener 2 valores: 0 o 1, que corresponde a 0 voltios o 3 voltios.
- Como salida digital, que solo puede tener 2 valores: 0 o 1, que coloca a la salida 0 voltios o 3 voltios.

Cada pin se configura automáticamente en función de la instrucción o bloque que se utilice. Si se usa una instrucción de entrada, ver Figura 10, quedará configurado como entrada.

Ahora, trabajen como equipo en la resolución del primer reto.

Reto 1.

Lean la situación a solucionar.



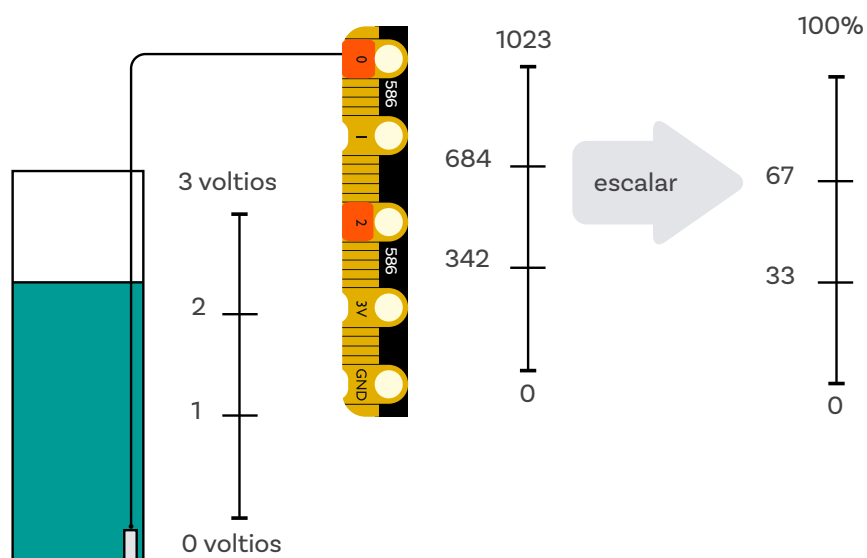
En tu escuela se ha instalado un tanque que recolecta el agua lluvia para aprovecharla en los baños y lavamanos. Para poder monitorear la cantidad de agua recolectada, se ha instalado un sensor de nivel de agua en el fondo del tanque. Este sensor se ha conectado al pin 0 de una micro:bit.

Si el sensor entrega 0 voltios, esto indica que el tanque está vacío, y si entrega 3 voltios, el tanque está lleno. Si, por ejemplo, el tanque se encuentra lleno solo hasta la mitad, el sensor colocará 1,5 voltios en el pin 0.

Sin embargo, estos valores en voltios no son claros para todas las personas. Por eso se requiere crear un programa que muestre el porcentaje de llenado del tanque en una escala de 0 a 100%. Su misión es crear este programa.

Antes de empezar, miren el diagrama que se presenta en la Figura 11.

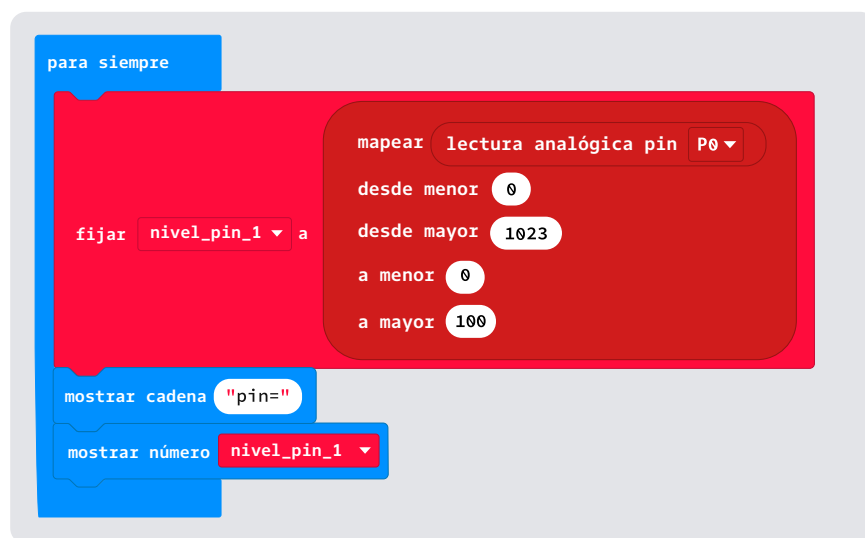
Figura 11. Diagrama Reto 1



Como se observa en la *Figura 11*, el sensor que se coloca al fondo del tanque mide el agua que hay encima de él y da una señal que varía de los 0 voltios a los 3 voltios. Al conectar este sensor al pin 0 de la *micro:bit* se lee un número entre 0 y 1023, que en este caso es el 586. Este valor indica que el tanque está algo más de medio lleno. Así que el reto es convertir ese valor en algo que se pueda mostrar en pantalla y que vaya en porcentajes de 0 a 100, que es una escala más usual.

Examinen ahora el programa de la *Figura 12* y utilicen el comentario agregado para entender su funcionamiento.

Figura 12. Código con bloque del menú pines llamado map



¿Qué hace?

Ahora prueben este programa y jueguen con el valor de entrada al pin 0, esperando cada vez un momento hasta que se muestre el porcentaje en la pantalla.



¿Podrían hacer que salga el valor seguido del símbolo %?

Si ya lograron lo anterior, resuelvan este segundo reto.

Reto 2

Lean la situación problema.



El código anterior se utilizó también para programar el tanque de recolección de aguas lluvias que se utiliza para regar las plantas en un invernadero. Sin embargo, se debe ajustar el programa para que avise cuando quede menos del 25% de agua en el tanque, a fin de llenarlo con agua de un pozo.

Su misión es hacer un programa que muestre en la pantalla un ícono que indique que el tanque tiene suficiente agua y otro ícono que indique que es hora de buscar otra fuente de agua, dado que el agua lluvia no es suficiente. Utilicen su creatividad para diseñar estos dos íconos.

Recuerden terminar con tiempo para que logren reflexionar y organizar las ideas sobre lo que aprendieron resolviendo los retos.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

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

1 ¿Puedes leer y escribir datos de dispositivos externos usando los pines?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

2 ¿Puedes cambiar de escala de las lecturas a una escala apropiada a la naturaleza de la variable?

- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

Si tus respuestas fueron “Parcialmente” o “Aún no”, vuelve a las actividades propuestas en MakeCode. Luego, discute con algunas compañeras y compañeros de tu clase lo que hace el código de MakeCode que se presenta en las diferentes figuras. Si todavía te quedan dudas, consúltale a tu docente.

Ahora piensa en lo siguiente:



Si regresas al reto del invernadero automatizado, ¿cómo te sirve lo que aprendiste en esta sesión?

Si en lugar de querer mostrar un ícono que indique que se debe llenar el tanque, lo que se quiere es encender una bomba que envíe agua al tanque, ¿cómo te imaginas que podría hacerse esto? Haz un dibujo con tu idea.

Aprovecha este espacio final para hacerte un esquema en que resumas algo de lo que aprendiste, indicando para qué sirven los pines y proponiendo un ejemplo de su uso y del ajuste de la escala de una señal.