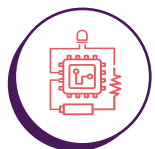


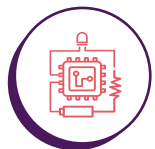
Sesión 4

Aprendizajes esperados

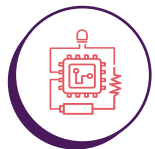
Al final de esta sesión verifica que puedas:



Aplicar lecturas digitales para producir cambios en salidas analógicas.



Aplicar cambios graduales en actuadores, usando la escritura analógica.

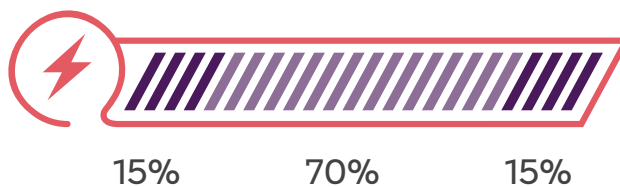


Conectar correctamente sensores externos a la *micro:bit*.

Material para la clase

- ☐ Tinkercad - cuenta de estudiante o personal.

Duración sugerida



Lo que sabemos,

lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

La seguridad es una prioridad en la ciudad del futuro, por lo que en esta sesión diseñaremos un sistema de alarma utilizando un **sensor de movimiento PIR**, un **piezo o parlante** y un LED. Este sistema de seguridad detectará intrusos y nos alertará con una alarma sonora. Al aprender sobre la variación de voltaje y la escritura analógica, podremos ajustar el tono y el volumen de la alarma, haciendo que nuestros hogares y edificios sean más seguros. Este proyecto demostrará cómo la tecnología puede protegernos y brindarnos tranquilidad en nuestra vida diaria.

En el mundo de la automatización, los sensores, actuadores y las lecturas y escrituras analógicas se integran para crear sistemas eficientes y precisos. Los sensores recopilan datos del entorno como la temperatura o la luminosidad y estos datos suelen ser señales analógicas que representan valores físicos variables. Las lecturas analógicas se convierten en datos digitales que la *Micro:bit* puede procesar, mientras que la escritura analógica se utiliza para controlar actuadores, como motores o luces LED, ajustando la energía suministrada de manera eficiente.

La **lectura analógica** se refiere a la capacidad de la *micro:bit* para medir voltajes variables en sus pines de entrada, a través de sensores. Estas lecturas permiten captar información del voltaje cambiante, a diferencia de las lecturas digitales, que solo detectan estados de encendido o apagado (HIGH o LOW). Por otro lado, la **escritura analógica** se refiere a la capacidad de la *micro:bit* para generar señales de salida de voltaje variable en sus pines, lo que permite modificar la acción de un actuador, como aumentar o disminuir la velocidad de un motor, el brillo de una bombilla o como en este proyecto cambiar el tono de sonido de un parlante o **piezo**.

Un ejemplo excelente de esta integración es un sistema de climatización inteligente en un hogar. Un sensor de temperatura mide continuamente la temperatura ambiente y estos datos se leen como señales analógicas, que luego se convierten en valores digitales para controlar la velocidad del ventilador

y la potencia del compresor en un aire acondicionado. Si la temperatura es muy alta, el sistema incrementa la velocidad del ventilador y la capacidad de enfriamiento del compresor para enfriar la habitación más rápidamente. Este sistema asegura un control preciso y eficiente del clima, manteniendo el confort y optimizando el consumo energético.

Glosario



Piezo o parlante: dispositivo que emite sonidos o tonos cuando recibe señales eléctricas, comúnmente utilizado para alertas o efectos sonoros en proyectos de programación.



Sensor PIR: dispositivo que detecta movimiento mediante cambios en la radiación infrarroja emitida por objetos en su entorno.



Lectura analógica: proceso de capturar valores continuos de un sensor, representados como un rango de números en lugar de sólo valores discretos como 1 o 0.



Escritura analógica: proceso de enviar señales de salida con valores continuos, simulando una variación gradual, como en el control de brillo o velocidad en lugar de solo valores discretos como encendido o apagado.



PWM: estrategia para modificar el valor de una señal variable en el tiempo, como el voltaje. Se utiliza, entre otras cosas, para controlar la velocidad en motores o el brillo en LED.

Figura 1. Selección de componentes en *Tinkercad*

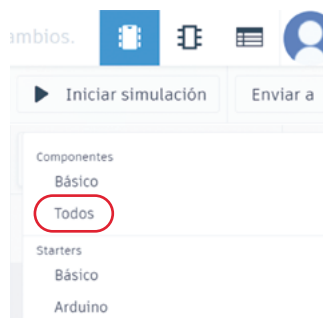


Figura 2. Conexión de LED a pin 0 de la *micro:bit*

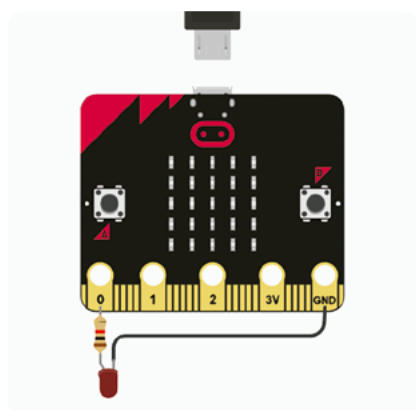
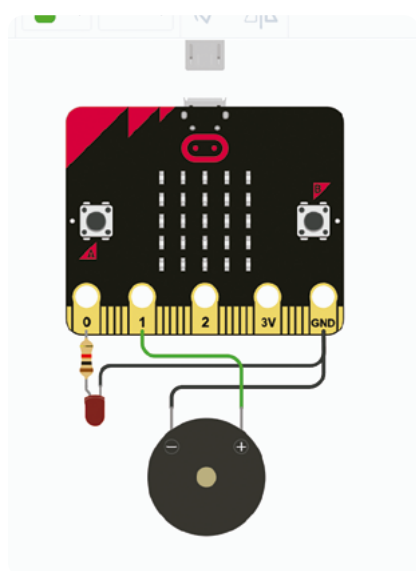


Figura 3. Conexión de piezo a la *micro:bit*



Manos a la obra

Conectadas



Esta sección corresponde al 85% de avance de la sesión

Para aplicar sensores y actuadores vamos a iniciar esta sesión con una práctica que simule el funcionamiento de una alarma que detecta el movimiento y emite un sonido que alerte la presencia. sigue los pasos:

Paso 1: crea un nuevo circuito desde *Tinkercad*, haz clic en la opción “componentes” y luego selecciona la opción “All” o “Todos”, *Figura 1*.

Paso 2: busca en la sección de componentes “output” o “salida” el LED e instálalo tal como en la práctica del semáforo, ver *Figura 2*.

No olvides la resistencia.

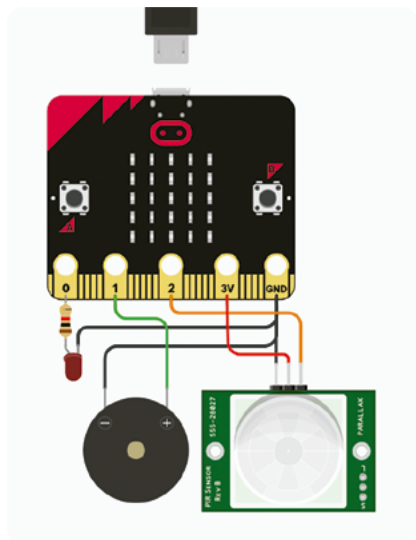
Paso 3: allí mismo en la sección “output” encontrarás un elemento llamado “piezo”, el cual es un dispositivo que funciona de forma semejante a un altavoz que emite sonidos de agudos a graves, dependiendo de la señal eléctrica que fluye por él. Después variaremos el sonido con una señal de voltaje variable. Instala el piezo, de acuerdo con la imagen, el polo negativo conectado a **GND** y el positivo a **pin 1**, ver *Figura 3*.

Paso 4: en la sección “input” o “entrada” de los componentes busca el **sensor PIR** o “**PIR sensor**”, que es un sensor digital que al detectar movimiento emite una señal de corriente eléctrica constante hasta que deja de detectar movimiento. Instálalo tal como se ve en la *Figura 4*.

El terminal “power” o “potencia” del sensor debe ir conectado a 3V, la terminal “ground” o “tierra” conectada a **GND** y “señal” o “señal” a **pin 2**.

Ahora comencemos con la programación.

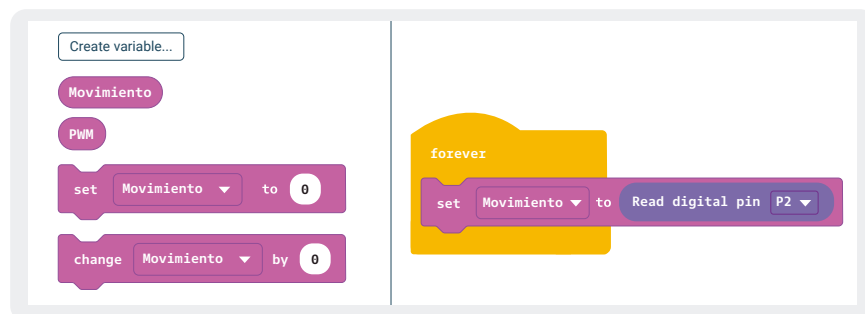
Figura 4. Conexión de Sensor PIR a la *micro:bit*



Paso 5: crea un par de variables llamadas “Movimiento” y “PWM”. Con la primera vamos a guardar el estado del sensor. Si se detecta movimiento tomará el valor de 1 y en caso contrario valdrá 0. La variable “PWM”, por otro lado, nos permitirá escuchar cómo cambia el sonido al variar el voltaje en el parlante o piezo.

Busca en la paleta “Input” la lectura del pin digital pin 2 y conéctalo a la variable “movimiento” dentro del ciclo “forever” o “Para siempre”, como se ve en la *Figura 5*.

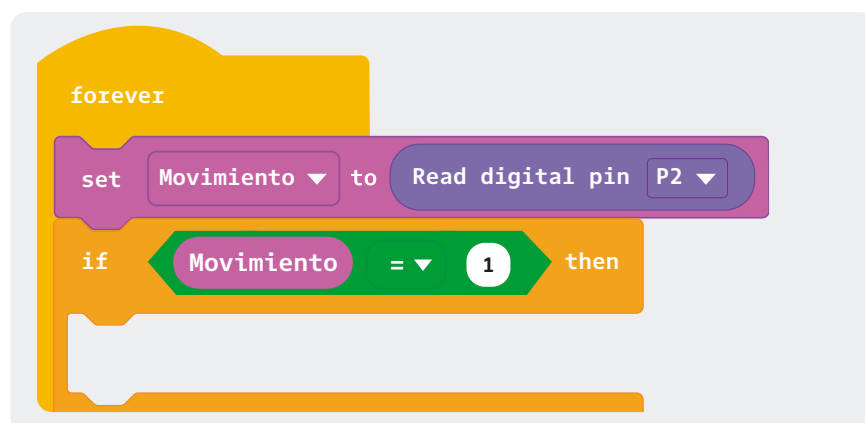
Figura 5. Creación y programación de variables



Ahora la variable “Movimiento” toma la lectura de pin 2, es decir la señal emitida por el sensor PIR.

Paso 6: desde la paleta “control” selecciona y conecta una condicional doble, luego desde la paleta “math” selecciona y arrastra a la pregunta de la condicional una igualdad y pregunta ¿La variable Movimiento es igual a 1?, ver *Figura 6*.

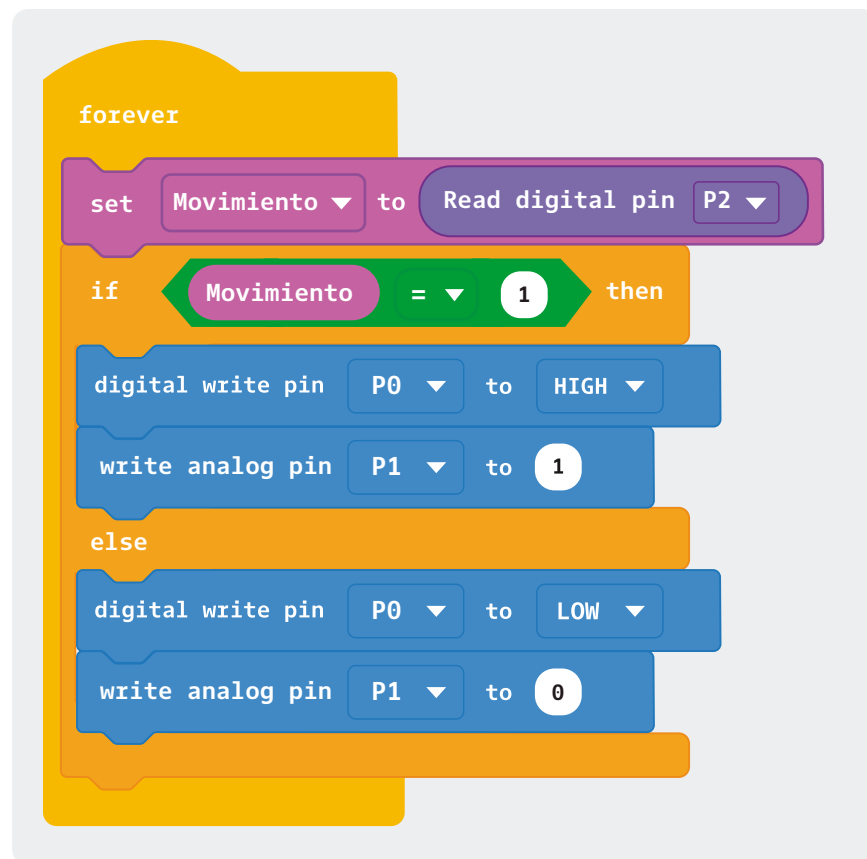
Figura 6. Programación condicional



En caso de que “Movimiento” sea igual a 1, quiere decir que el sensor PIR está detectando movimiento, así que encenderemos el LED conectado a “pin 0” pasándolo en una escritura digital a “HIGH”. Por otra parte, si la sentencia es verdadera también vamos a hacer sonar el piezo. Sin embargo, fíjate que para cambiar el sonido debemos modificar la señal eléctrica, para lo cual usaremos una escritura analógica que inicialmente vamos a dejar con un valor de 1.

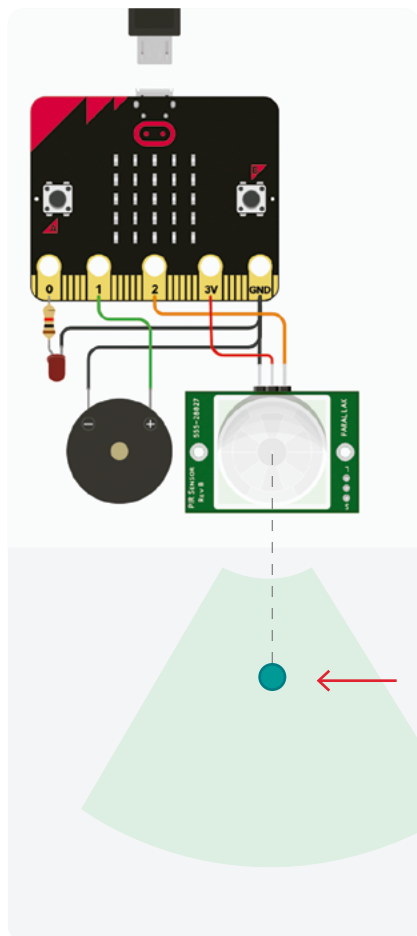
En caso de que “Movimiento” sea 0 el Led debe estar en “LOW” y el piezo en 0 como se ve en la *Figura 7*.

Figura 7. Programación para “Movimiento” en 0



Ahora vamos a usar la variable “PWM” para hacer que el sonido del piezo cambie cada que haya movimiento.

Figura 8. Acción de incremento de variable “PWM”



Paso 7: en la paleta variable selecciona y conecta en el lado verdadero de la condicional la opción “Change (PWM) by (1)” y donde estaba el 1 en la escritura analógica ensambla el valor de “PWM”. Esto hará que cada vez que el sensor detecte movimiento se incremente en 1 el valor de la variable “PWM” y al escribirse en la salida del piezo hace que cambie el sonido, ver *Figura 8*.

Reinicia el valor de “PWM” a cero en caso de que el sensor no detecte movimiento, como se ve en la *Figura 9*.

Figura 9. Establecer valor de PWM a cero



Paso 8: simula el montaje dando clic en “Start simulation” o “Iniciar simulación”.



Start Simulation

Mientras se ejecuta la simulación, haz clic sobre el sensor PIR y arrastra el círculo azul para simular el movimiento que detectará la alarma, como se observa en la *Figura 8*.



¿Notas el cambio de sonido al simular movimiento en el montaje?

La variable PWM sirve para ajustar el voltaje variable de la salida, aunque en este ejemplo lo usamos para modificar continuamente la señal y así provocar cambios en el sonido.



¿Cómo se te ocurre que puedes usar sensores y actuadores para solucionar un problema en tu contexto?

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

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

1

¿Puedes diferenciar entre entradas y salidas analógicas y digitales?



Sí



Parcialmente



Aún no

- 2 ¿Puedes hacer lecturas analógicas de sensores externos y controlar actuadores externos desde la escritura analógica?
- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no
- 3 ¿Puedes conectar sensores y actuadores externos a la *micro:bit* desde el simulador *Tinkercad*?
- ☐ Sí
- ☐ Parcialmente
- ☐ Aún no

Si tus respuestas fueron “Parcialmente” o “Aún no”, discute con tu compañera o compañero de grupo el trabajo realizado sobre sensores y actuadores, verifica la solución que dieron a uno de los ejercicios planteados y consulta con tu docente las preguntas que aún tengas.

A continuación te proponemos algunas preguntas para discutir en grupo y luego compartir:



Si regresamos al reto, ¿para qué serviría lo que hemos aprendido en esta sesión?

¿Qué ventajas tendría trabajar con lecturas y escrituras análogas frente a lecturas y escrituras digitales de encendido y apagado?

Aprovecha este espacio final para crear un esquema que sintetice lo aprendido, incorporando definiciones de términos clave y ejemplos relacionados con los conceptos aplicados al programar una alarma que, al detectar movimiento, ajusta el tono de un **piezo** mediante **escritura analógica** en la *micro:bit*.