

PRÁCTICAS MICRO:BIT

MISIÓN 2: EL PROBLEMA DEL RUIDO INVISIBLE



Índice

1. Misión 2: el problema del ruido invisible - 2 -
2. Programación - 4 -

1. MISIÓN 2: EL PROBLEMA DEL RUIDO INVISIBLE

Tras el éxito de la primera misión y la activación del prototipo **W-01**, el laboratorio del **Doctor Elio Wolfram** volvió a llenarse de actividad. Los sistemas básicos del instituto comenzaban a responder... pero un nuevo problema no tardó en aparecer.

Una tarde, mientras trabajábamos en el laboratorio, el Doctor Wolfram frunció el ceño mientras observaba unos gráficos en su ordenador.

— *“Algo no va bien”* —murmuró—.

Nos explicó que había detectado un **nivel de ruido ambiental demasiado alto** en varias aulas del instituto. Ese ruido constante dificultaba las explicaciones del profesorado, impedía la concentración y aumentaba el **estrés del alumnado**, incluso sin que nos diéramos cuenta.

“El ruido no siempre se ve, pero sus efectos son muy reales” —nos advirtió.

El desafío del sonido

El Doctor Wolfram había tenido una idea: crear un **sistema automático de control del ruido** que ayudara a mantener un ambiente adecuado para el aprendizaje.

— *“No se trata de castigar”* —nos explicó— *“sino de avisar cuando el sonido supera un nivel razonable para una clase.”*

El sistema debía funcionar de la siguiente manera:

- Medir el **nivel de sonido ambiental**
- Compararlo con un **valor máximo recomendado**
- Activar una **alarma** cuando el ruido fuera excesivo

Pero para conseguirlo, necesitaba de nuevo nuestra ayuda.

La tecnología del W-01

El Doctor Wolfram nos recordó que el prototipo **W-01** cuenta con una **placa micro:bit V2**, equipada con un **micrófono integrado**, capaz de detectar el nivel de sonido del entorno.

— “La *micro:bit* puede oír lo que ocurre en el aula” —dijo sonriendo—. “Solo necesita que alguien le diga cuándo el ruido es demasiado alto.”

Y ese alguien éramos nosotros.

El objetivo de la misión

Nuestra misión será **programar la *micro:bit*** para que:

- Detecte el nivel de sonido del aula
- Compare el sonido con un **umbral máximo recomendado para una clase**
- Active una **alarma visual y/o sonora** cuando se supere ese nivel

El Doctor Wolfram nos indicó que, para un entorno de clase adecuado, se considerará **ruido alto** cuando el nivel de sonido supere aproximadamente:

65 decibelios (dB) (*equivalente a una conversación fuerte y continua*)

Preparados para programar

— “Usaremos de nuevo **Microsoft MakeCode**” —nos dijo el Doctor—, “para enseñar al *W-01* a escuchar, decidir y avisar.”

Encendimos nuestros ordenadores sabiendo que esta vez no solo íbamos a programar un robot, sino a **mejorar el ambiente del instituto**.

Antes de comenzar, el Doctor Wolfram nos dejó un último mensaje:

“Un buen ingeniero no solo crea tecnología... también piensa en las personas que la van a usar.”

La **Misión 2** acababa de comenzar.

2. PROGRAMACIÓN

Accedemos a **Microsoft MakeCode** (<https://makecode.microbit.org/>) y generamos un *Nuevo proyecto*, al que llamaremos *Misión 2: Ruido invisible*.

Tal y como nos ha comentado el Dr. Wolfram, la tarjeta micro:bit V2 incorpora un micrófono capaz de registrar el nivel de sonido ambiental del aula.

Debemos establecer un valor umbral por encima del cual consideraremos que hemos sobrepasado el límite de ruido ambiental en la clase. En ingeniería, el nivel de sonido se mide en una unidad un poco especial que se denomina decibelio (dB). No vamos a adentrarnos en detalles técnicos, simplemente vamos a considerar que el nivel de sonido máximo admisible para que una clase se desarrolle con normalidad es de 65 dB. Este valor se corresponde con el de una conversación suave.

Sin embargo, el Dr. Wolfram nos alerta de un pequeño inconveniente: el sensor de sonido de la micro:bit V2 mide niveles de sonido del 0 (silencio) al 255 (máximo), no directamente en decibelios (dB). Para solucionarlo, podemos considerar la siguiente equivalencia:

- **Silencio (0 dB):** Lo más cercano a 0 en la micro:bit (o un valor muy bajo, como 20-30) se corresponde con el silencio.
- **Conversación (~65 dB):** En un entorno tranquilo, un valor de lectura alrededor de 100-150 en la micro:bit podría corresponder con una conversación normal (aprox. 65 dB).
- **Ruido Fuerte (> 70 dB):** Sonidos como tráfico intenso (70-80 dB) darían valores cercanos a 200 o más en el sensor de la micro:bit.

El Dr. Wolfram nos resume todo esto de una manera muy sencilla: *“Deberemos fijar el límite de ruido en un valor entre 100 y 150 de la escala que utiliza nuestra micro:bit. Para ajustar bien el valor, podemos realizar varias pruebas hablando con un volumen adecuado para una clase y observando donde debería estar el límite”*.

Para trasladar este código a *MakeCode* deberemos usar un bloque condicional (**condición: si el ruido es mayor a 100 entonces suena una alarma**). Esta comprobación la debe realizar continuamente, por lo tanto, la incluiremos dentro del bloque *“Para siempre”*:

Buscamos dentro del menú de *MakeCode* el apartado **Lógica** y arrastramos el primer bloque condicional dentro del “*para siempre*”.

A continuación, establecemos la condición: “*nivel de sonido*” > 100:

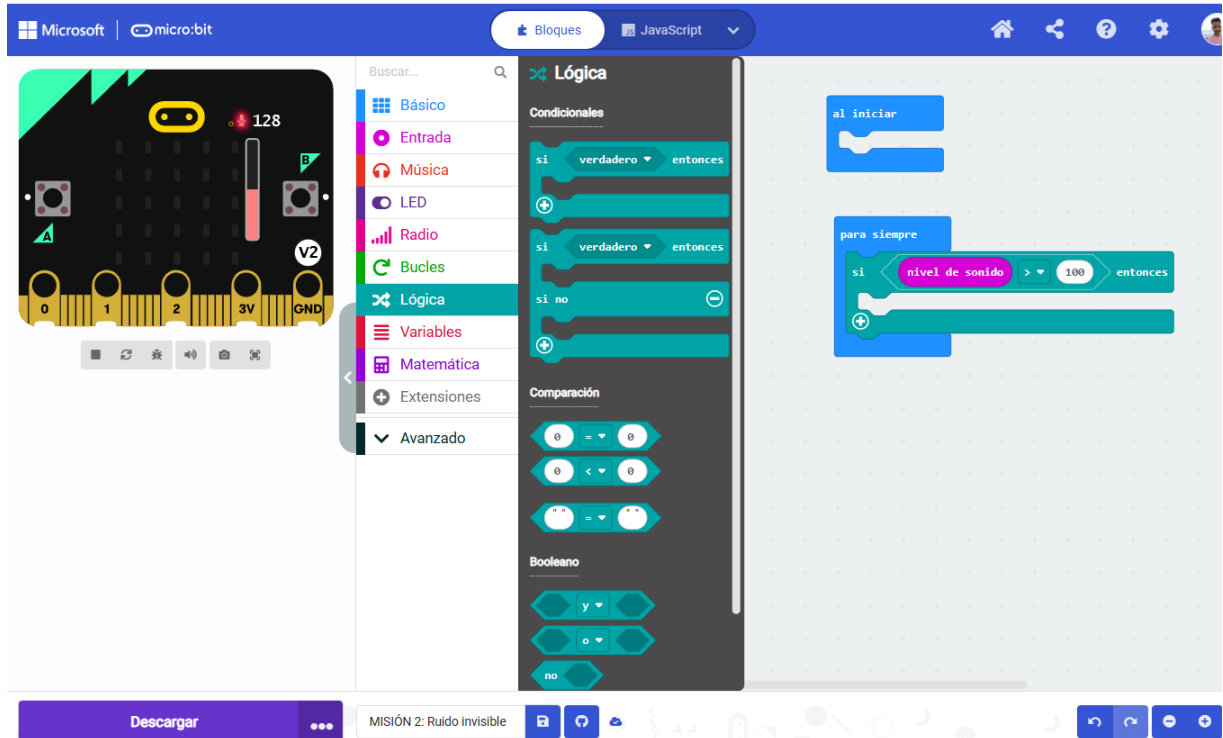


Figura 1. Condición para activar el aviso de ruido en el aula

Ahora debemos decidir la acción a tomar cuando se supere el umbral de ruido. Acordamos con el Dr. Wolfram que la tarjeta micro:bit debería emitir una señal sonora de aviso mientras se esté superando el límite establecido y, además, mostrar una señal luminosa a modo de cruz roja en la matriz de LEDs de la tarjeta programable.

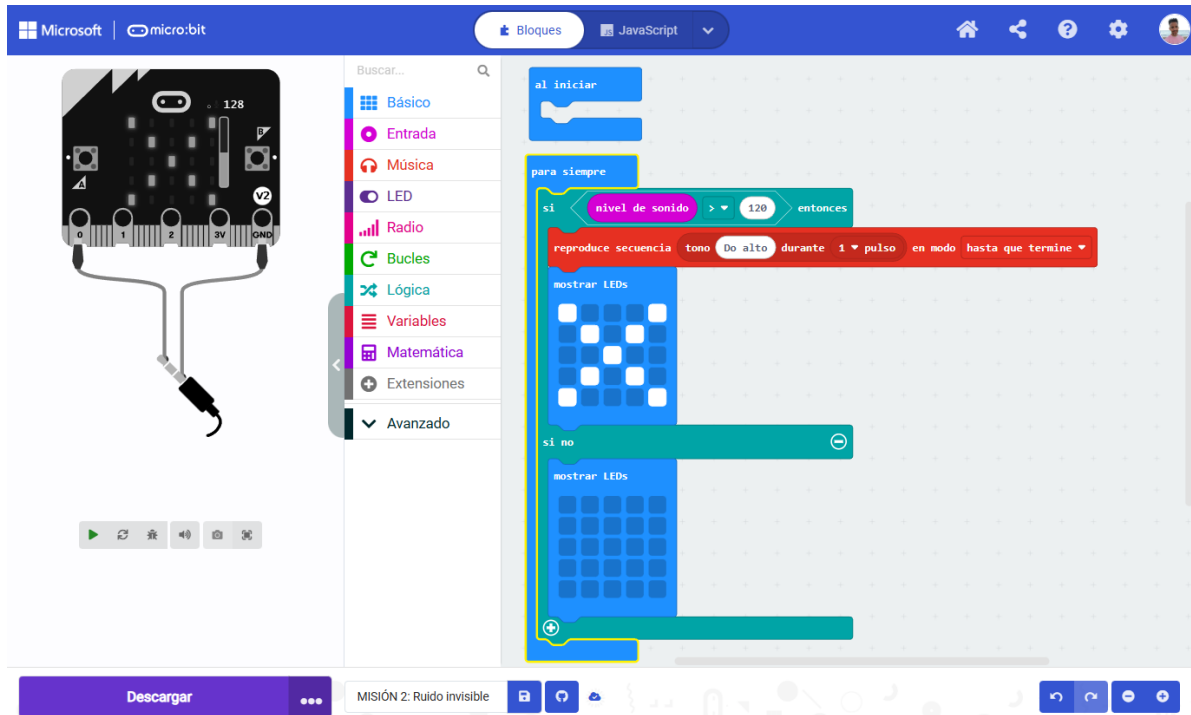


Figura 2. Código para que emita sonido con ruido alto y muestre señal luminosa

¡Ya casi lo tenemos! – dice el Dr. Wolfram -. Descarga el programa en tu micro:bit y realiza varias pruebas para comprobar que el límite de 100 que hemos establecido es correcto, o quizás debamos modificarlo ligeramente.

Tras varias pruebas, a 100, 150, 120, observamos que 120 es un valor adecuado para limitar el ruido ambiental del aula.

Pero esto no es todo, el Dr. Wolfram nos propone mejorar aún más el código incluyendo un **gráfico de barras en la matriz de LEDs** de la micro:bit, para que vaya mostrando el nivel sonoro del aula. Además, activaremos el **botón A** de la tarjeta para que, al pulsarlo, nos muestre el mayor nivel de ruido que ha registrado el micrófono hasta ese momento. Veamos cómo podemos hacerlo.

- Vamos al menú de **“Variables”** y creamos una variable que se llame **MaxSound** (sonido máximo). En esta variable iremos almacenando el mayor valor ruido registrado hasta ese momento. Dentro del bloque **“al iniciar”** fijamos el valor de la variable **MaxSound** a cero, para inicializar el programa.

- Dentro del bloque **“para siempre”**, incluimos otro bloque condicional para establecer lo que sucede cuando pulsamos el *botón A*. Si pulsamos el *botón A*, la *micro:bit* debe mostrar la cadena *MaxSound* (valor de ruido máximo).
- Si no, creamos una nueva variable que se llame *SoundLevel* (nivel sonoro) y fijamos su valor a nivel de sonido. A continuación, trazamos el gráfico de barras con los valores de *SoundLevel*. El gráfico de barras lo encontramos dentro del menú **LED**.
- Para terminar, incluimos un último bloque condicional dentro del anterior, en el que comprobamos si la variable *SoundLevel* es mayor que *MaxSound*. Si se cumple la condición, fijamos en *MaxSound* el valor de la variable *SoundLevel*, para actualizar su valor.

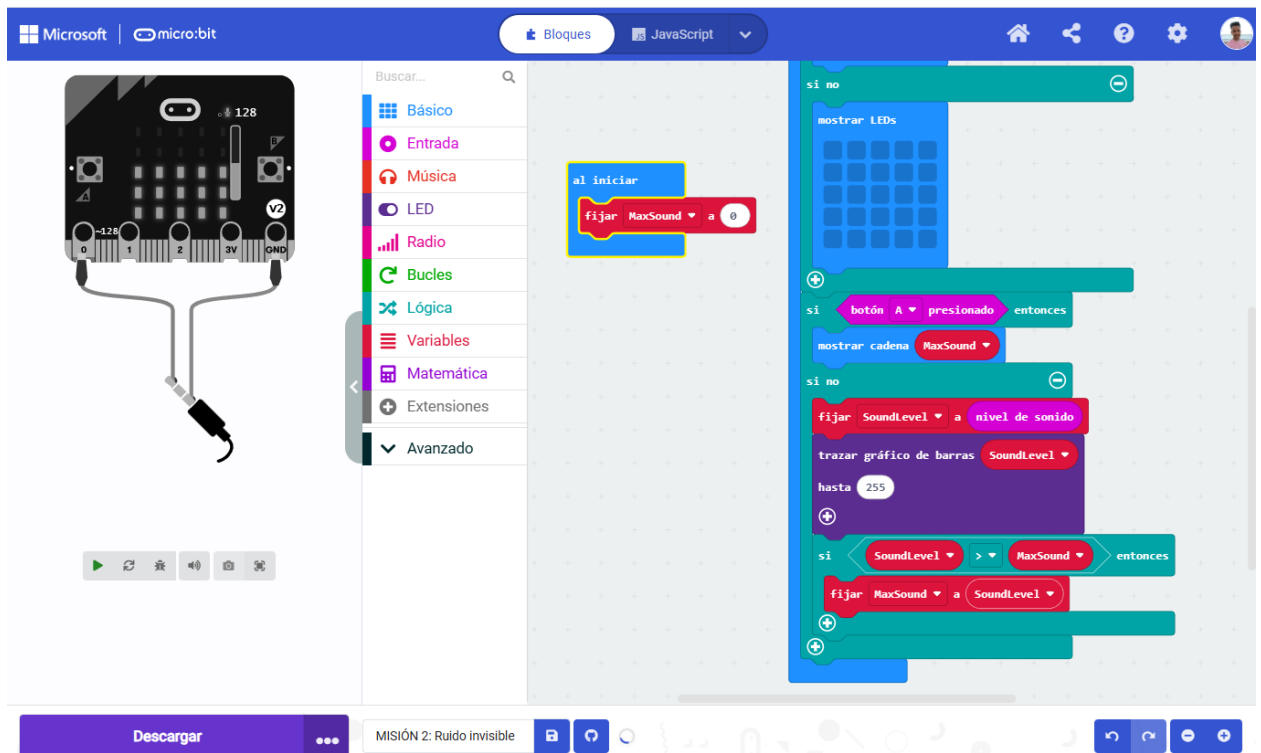


Figura 3. Código completo

Comprobamos que el programa funciona correctamente y hace exactamente lo que esperábamos, según las indicaciones del Dr. Wolfram.

Mensaje final del Dr. Wolfram:

Fin de la Misión 2

“Excelente trabajo, jóvenes ingenieros e ingenieras del IES Hermanos D’Elhuyar.

Habéis logrado algo muy importante: enseñar al prototipo W-01 a escuchar su entorno y a reaccionar cuando el ruido supera un nivel adecuado para aprender.

Gracias a vuestro programa, ahora el aula puede avisarnos cuando el sonido es demasiado alto, ayudándonos a crear un ambiente más tranquilo, respetuoso y favorable para el estudio.

Esto demuestra que la tecnología no solo sirve para construir máquinas, sino también para mejorar la convivencia y el bienestar de las personas.

Seguid observando, pensando y programando. Pronto necesitaré vuestra ayuda para una nueva misión aún más desafiante.

Enhorabuena por completar con éxito la Misión 2: Sistema de alarma por ruido ambiental.”

🏆 Insignia obtenida: Ingeniero/a de Sonido del Aula