

Monitoreando el ruido en el aula

Grado sugerido: Décimo

Julio Cesar Herrera Marin

Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.

Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](#)

Para contactar al autor/a de este recurso, escriba a: Julio.herrera@eleazarlibreros.edu.co

PLANTILLA DE PROYECTO

EL GUARDIÁN DEL SILENCIO: MONITOREANDO EL RUIDO EN EL AULA CON MICRO:BIT

Duración	3 sesiones de 90 minutos cada una (Total: 270 minutos)
Objetivo y descripción del proyecto	<i>En este proyecto, los estudiantes diseñarán y programarán un dispositivo usando la tarjeta micro:bit para monitorear el nivel de ruido en el aula y mostrar si el ambiente es "Silencioso", "Normal" o "Ruidoso" mediante su matriz de LEDs. A través de este ejercicio, los estudiantes aplicarán principios de pensamiento computacional como la descomposición de problemas, el diseño de algoritmos para toma de decisiones, la programación basada en condiciones, la depuración y el uso de sensores en un contexto real.</i>
Lista de materiales	<ul style="list-style-type: none">• Una tarjeta micro:bit por estudiante o grupo (v2 recomendada por su micrófono integrado).• Un cable USB para conectar la micro:bit a la computadora.• Una computadora con acceso a internet para usar el editor en línea MakeCode.• (Opcional, para visualización adicional) Unos pocos LEDs y resistencias, y protoboard pequeña.
Características del problema para tener en cuenta en la solución.	<ul style="list-style-type: none">• Detección de ruido: La micro:bit debe ser capaz de "escuchar" el sonido ambiente. La micro:bit v2 tiene un micrófono integrado que simplifica esto.• Umbrales de ruido: Es necesario definir umbrales claros para clasificar el ruido en "Silencioso", "Normal"

	<p>y "Ruidoso". Estos umbrales pueden necesitar calibración inicial en el aula real.</p> <ul style="list-style-type: none"> • Visualización: La micro:bit debe mostrar el estado del ruido de manera clara y comprensible para los estudiantes. La matriz de LEDs es ideal para esto (ej. caras felices/tristes, íconos de volumen). • Interacción y retroalimentación: El sistema debe proporcionar una retroalimentación inmediata sobre el nivel de ruido. • Portabilidad: El dispositivo debe ser lo suficientemente portátil para ser ubicado en diferentes puntos del aula.
Pasos para desarrollar el proyecto	<p>Sesión 1: Entendiendo la micro:bit y el Sonido (90 minutos)</p> <p>1. Introducción a micro:bit y MakeCode (15 minutos):</p> <ul style="list-style-type: none"> • Explicar qué es la micro:bit y sus componentes básicos (matriz de LEDs, botones, sensor de luz, micrófono - v2). • Introducir el entorno de programación MakeCode (editor por bloques). • Cargar un programa simple de "Hola Mundo" en la micro:bit para familiarizarse con el proceso. • Concepto de computación:: Abstracción (entender la micro:bit como un dispositivo que simplifica la electrónica), Programación (primer programa simple). <p>2. Explorando el Sensor de Sonido (45 minutos):</p> <ul style="list-style-type: none"> ▪ Lectura del sensor de sonido en MakeCode: <ul style="list-style-type: none"> ▪ Explicar cómo usar el bloque "nivel de sonido" o "leer pin analógico" (para sensor externo) en MakeCode.

	<ul style="list-style-type: none"> ▪ <i>Pedir a los estudiantes que muestren el valor de la lectura del micrófono/sensor en la matriz de LEDs o en la consola del simulador.</i> ▪ <i>Realizar pruebas: hablar en voz baja, aplaudir, gritar, y observar cómo cambian los valores.</i> ▪ Concepto de Computación: <i>Recopilación y análisis de datos (observar y entender los valores del sensor), Lógica (cómo se interpretan esos valores).</i> <p>3. Definiendo Umbrales de Ruido (30 minutos):</p> <ul style="list-style-type: none"> • <i>En grupos, los estudiantes deben experimentar con el micrófono y definir 3 rangos de valores de sonido que representen "Silencioso", "Normal" y "Ruidoso" en su aula.</i> • <i>Anímelos a registrar estos valores y justificarlos (ej. "menos de 50 es silencioso porque...").</i> • Concepto de Computación: <i>Modelado y simulación (al establecer umbrales basados en la experimentación), Abstracción (simplificar la complejidad del sonido en rangos).</i> <p>Sesión 2: Programando la Lógica del Guardián (90 minutos)</p> <p>1. Diseño del Algoritmo (30 minutos):</p> <ul style="list-style-type: none"> ○ <i>En sus cuadernos, pídeles que diseñen un algoritmo (en pseudocódigo o diagrama de flujo) que, basándose en los umbrales definidos, decida qué icono mostrar en la micro:bit.</i> ○ <i>Debe incluir estructuras condicionales (sí...entonces...sino sí...sino)</i> ○ <i>Ejemplo de pseudocódigo:</i> ○ <i>INICIO</i> <i>REPITE SIEMPRE:</i> <i>LEER nivel_sonido_microfono</i> <i>SI nivel_sonido_microfono <</i> <i>UMBRAL_SILENCIO ENTONCES:</i>
--	---

MOSTRAR icono_silencio (ej. cara feliz)
 SINO SI nivel_sonido_microfono <
 UMBRAL_NORMAL ENTONCES:
 MOSTRAR icono_normal (ej. cara neutral)
 SINO:
 MOSTRAR icono_ruidoso (ej. cara triste/X)
 FIN REPETIR
 FIN

- **Concepto de Computación:** Diseño de algoritmos, lógica, descomposición (dividir el problema en lectura, decisión y visualización).

2. **Codificación en MakeCode (60 minutos):**

- Traducir el algoritmo diseñado a bloques de MakeCode.
- Utilizar los bloques "si... entonces... sino si... sino" (if/else if/else) para implementar la lógica de los umbrales de ruido.
- Usar los bloques "mostrar icono" o "mostrar LEDs" para representar los estados de ruido.
- Cargar el código a la micro:bit y probarlo en el aula.
- **Concepto de Computación:** Programación, depuración (identificar y corregir errores en el código o los umbrales), algoritmos.

Sesión 3: Pruebas, Calibración y Mejoras (90 minutos)

1. **Calibración y Depuración (45 minutos):**

- Los estudiantes deben probar el dispositivo en diferentes momentos del día y en diferentes ubicaciones del aula.
- ¿Funcionan bien los umbrales? ¿Los iconos reflejan correctamente el nivel de ruido percibido?

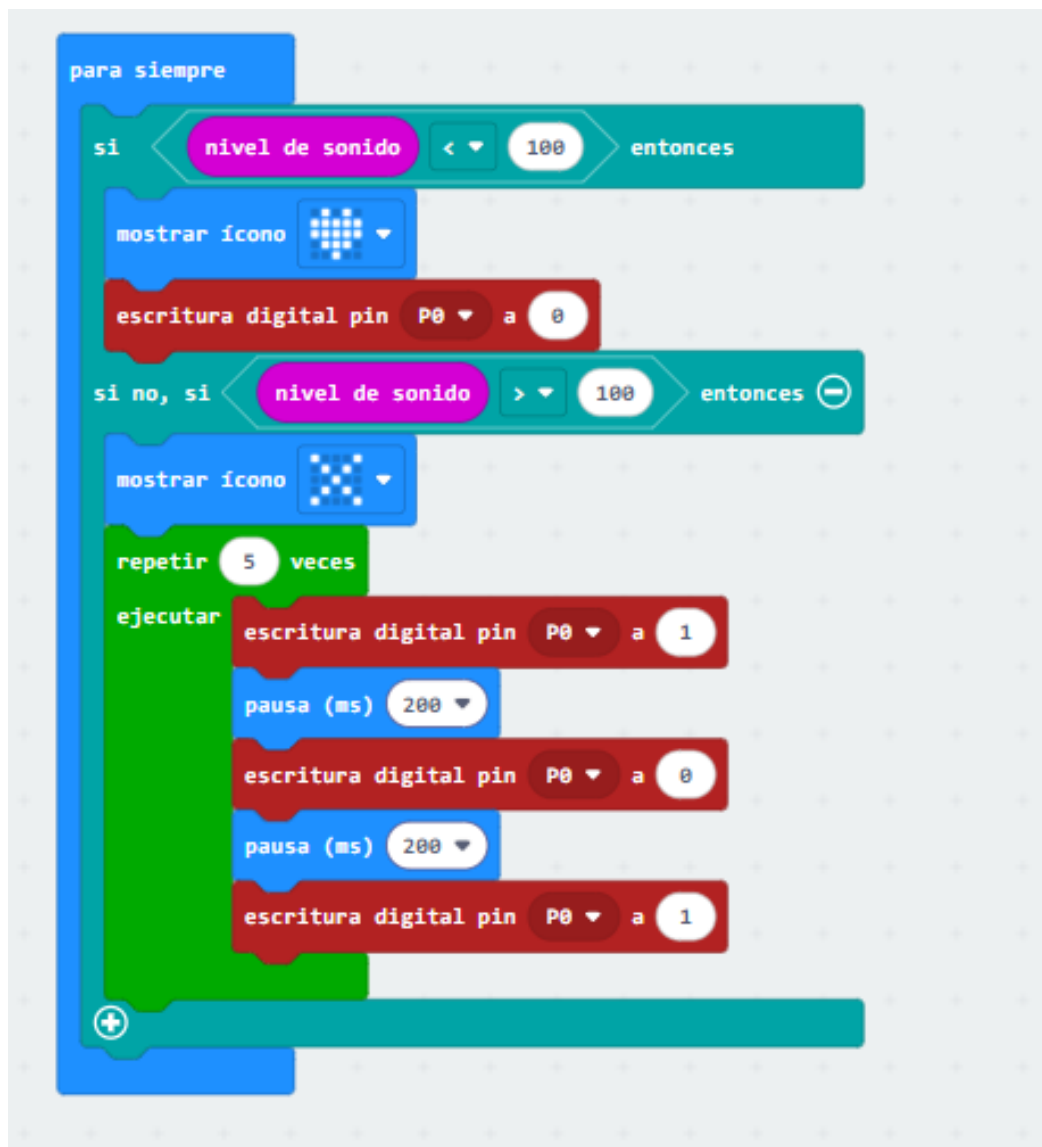
	<ul style="list-style-type: none"> • <i>Realizar ajustes finos en los valores de los umbrales en el código MakeCode.</i> • Concepto de Computación: <i>Depuración, prácticas de datos (ajustar el modelo con datos reales), modelado y simulación.</i> <p>2. Reflexión y Posibles Mejoras (30 minutos):</p> <ul style="list-style-type: none"> • <i>Realizar una sesión de lluvia de ideas sobre cómo mejorar el "Guardián del Silencio".</i> • <i>Posibles mejoras:</i> • <i>Incorporar un botón para "calibrar" (establecer nuevos umbrales).</i> • <i>Mostrar el valor numérico del sonido además del icono.</i> • <i>Registrar datos de ruido a lo largo del tiempo.</i> • <i>Conectar un altavoz para emitir una alarma si el ruido es excesivo.</i> <p>3. Presentación del Proyecto (15 minutos):</p> <ul style="list-style-type: none"> • <i>Cada grupo presenta su "Guardián del Silencio", explica su funcionamiento, los desafíos encontrados y las mejoras futuras.</i>
Adaptaciones	<ul style="list-style-type: none"> • Sin acceso a Internet: <i>Descargar el editor de MakeCode offline para trabajar sin conexión. Los programas se pueden guardar en archivos .hex y transferir a la micro:bit mediante USB. Las pruebas de calibración se pueden hacer con grabadoras de voz y análisis manual de los niveles de decibelios si no hay micrófono integrado.</i> • Población con discapacidad auditiva: <i>Además de la visualización en la matriz de LEDs, el dispositivo podría incluir una salida vibratoria (si se conecta un motor de vibración) o una señal luminosa externa (LED de color) para indicar los niveles de ruido.</i>

	<ul style="list-style-type: none"> • Población con discapacidad visual: El dispositivo podría integrar un altavoz para emitir sonidos o mensajes de voz pregrabados que indiquen el nivel de ruido (ej. "Ruido alto", "Ruido normal"). La programación en MakeCode es visual, pero se podría proporcionar código ya hecho o asistencia verbal para la construcción de bloques. • Recursos limitados (pocas micro:bit): Trabajar en grupos más grandes para compartir los recursos. Enfatizar la fase de diseño de algoritmo y pseudocódigo antes de pasar a la implementación física.
Referencias	<ul style="list-style-type: none"> • Fundación Raspberry Pi. (n.d.). MakeCode Projects with micro:bit. Recuperado de https://makecode.microbit.org/ • Micro:bit Educational Foundation. (n.d.). Official micro:bit website. Recuperado de https://microbit.org/.

ANEXO(s)

Anexo 1: Capturas de Pantalla de Bloques MakeCode. Imágenes del código MakeCode que muestren la lógica principal de lectura del sensor y condicionales

Ejemplo de código de bloques, en este caso se manejan dos lecturas y cuando el sonido es critico enviar señales a los leds



- **Anexo 2: Plantilla de Registro de Calibración.** Una tabla simple para que los estudiantes registren los valores de sonido en diferentes entornos (silencioso, normal, ruidoso) durante la fase de experimentación.

PLANTILLA DE VALORES DE RUIDO			
Situación del Aula	Valores Mínimos de Sonido (Ej. 0-255)	Valores Máximos de Sonido (Ej. 0-255)	Observaciones (Ej. "todos en silencio", "conversación normal", "hora del recreo")
Prueba 1: Silencio			
Prueba 2: Conversación baja			
Prueba 3: Ruido ambiente normal			
Prueba 4: Ruido moderado			
Prueba 5: Ruido alto / Gritos			
Prueba 6: (Otra situación)			

Anexo 3: Sitio web de Makecode con código de ejemplo

<https://www.microbit.org/es-es/projects/make-it-code-it/sound-meter/>