

Alerta Guardián

Etapa: Educación Secundaria Obligatoria	Curso: 1º-2º	Dificultad: baja
Temporalización: 90 minutos	Sesiones: 2	Áreas: Tecnología y Digitalización
Palabras clave: alarma, seguridad, sensores, programación, micro:bit, ultrasonidos, automatización		

1. Introducción

El mundo se enfrenta a una crisis inesperada: una **invasión zombi** amenaza con poner en peligro a los pocos supervivientes que quedan. Para proteger los refugios seguros, necesitamos diseñar un sistema que nos alerte inmediatamente si un zombi se acerca.

El reto que se plantea consiste en **construir un sistema de alarma automático** capaz de **detectar presencias** a una determinada distancia y **activar una señal sonora o visual** de advertencia.

Para lograrlo, utilizaremos una placa **micro:bit**, la placa del robot Maqueen, un **sensor ultrasónico de distancia** y actuadores incluidos en estos dispositivos como **LEDs** o **buzzers**, que programaremos mediante el entorno **MakeCode**.

Durante esta actividad, el alumnado trabajará en equipo para diseñar, montar, programar y poner a prueba su propio **sistema de alarma anti-zombis**, combinando habilidades de diseño tecnológico, pensamiento computacional, programación básica y cooperación en proyectos prácticos.

¿Serás capaz de construir un sistema fiable que detecte a los zombis antes de que sea demasiado tarde?

2. Marco curricular

2.1. Objetivos generales

- Diseñar y construir un sistema de alarma automatizado que funcione con sensores y microcontroladores.
- Programar la respuesta de un sistema técnico ante un estímulo físico (presencia/detector).
- Aplicar conocimientos de programación, electrónica y control automatizado en un reto contextualizado.
- Fomentar el trabajo cooperativo, la resolución de problemas y la toma de decisiones técnicas.

2.2. Competencias clave (LOMLOE)

Competencia STEM

- Aplicación de principios físicos para la detección de objetos.
- Resolución de problemas mediante el uso de sensores.
- Desarrollo de soluciones tecnológicas para necesidades concretas.

Competencia digital

- Programación de sistemas electrónicos simples.
- Configuración de sensores para la detección automatizada.
- Integración de componentes en un sistema funcional.

Competencia personal, social y de aprender a aprender

- Análisis de problemáticas de seguridad y sus soluciones.
- Desarrollo de proyectos tecnológicos con aplicación práctica.
- Reflexión sobre el proceso de creación tecnológica.

2.3. Competencias específicas, criterios de evaluación y saberes básicos

Área: Tecnología y Digitalización

Competencias específicas	Criterios de evaluación	Saberes básicos
CE1. Identificar necesidades tecnológicas del entorno.	C1.1 Definir problemas técnicos y buscar información fiable.	Bloque A: Estrategias de resolución de problemas.
CE2. Abordar problemas tecnológicos con creatividad y autonomía.	C2.2 Planificar y organizar recursos para construir soluciones técnicas.	Bloque A: Proceso tecnológico; materiales y herramientas.
CE5. Desarrollar algoritmos con pensamiento computacional.	C5.1 Diseñar y depurar programas para controlar sistemas automáticos.	Bloque B y C: Programación secuencial y sensores.
CE6. Comprender sistemas de control programado.	C6.1 Montar y programar sistemas automáticos con sensores y actuadores.	Bloque C: Fundamentos de robótica; control automático.

3. Planificación didáctica

3.1. Conocimientos Previos

- Conocimiento básico de estructuras de programación en MakeCode.
- Uso de micro:bit y sensores.
- Montaje de componentes electrónicos simples.
- Reparto de roles en el trabajo en equipo.

3.2. Metodología

La actividad se desarrolla bajo la metodología principal del **Aprendizaje Basado en Proyectos (ABP)**, donde el alumnado debe resolver un reto realista y contextualizado.

Los estudiantes trabajan por equipos para diseñar, programar y montar un sistema funcional que detecte una presencia y active una alarma. A lo largo del proceso, deben aplicar conocimiento técnico, colaborar, planificar tareas y tomar decisiones fundamentadas.

Estrategias didácticas complementarias:

- Trabajo cooperativo.
- Aprendizaje manipulativo y por experimentación.
- Pensamiento computacional.
- Resolución de problemas.
- Iteración, depuración y mejora.

El docente actúa como facilitador del proceso.

3.3. Temporalización, Espacios, Materiales y Recursos

Temporalización:

- Sesiones: 2
- Duración por sesión: 45 minutos

Sesión 1: Diseño y montaje

- Presentación del reto (5 min)
- Formación de equipos y asignación de tareas (5 min)
- Explicación del sensor ultrasónico y prueba (10 min)
- Montaje del sistema de detección (15 min)
- Programación inicial (10 min)

Sesión 2: Programación y prueba final

- Revisión del montaje y código (5 min)
- Mejora del algoritmo y ajustes de distancia (20 min)
- Prueba final y demostración (10 min)
- Relleno del diario de aprendizaje (10 min)

Espacios:

- Aula de Tecnología o aula de Informática.
- Distribución en parejas o pequeños grupos

Materiales y recursos:

- Placa micro:bit
- Sensor de distancia (HC-SR04 o similar)
- Zumbador o LED
- Ordenador con acceso a MakeCode
- Cables y placa de conexión (si se requiere)

3.4. Desarrollo Técnico por Sesiones

Esta actividad se llevará a cabo en **dos sesiones de 45 minutos** cada una. A continuación, se detalla la distribución del tiempo y las fases de trabajo técnico:

- **Sesión 1 (45 minutos)**

Objetivo: Comprender el reto, diseñar el sistema de alarma y comenzar la programación básica.

1. Presentación de la actividad (10 minutos)

- Breve explicación del reto: diseñar un sistema de alarma de presencia con sensor de distancia y alerta sonora o visual.
- Recordatorio de los conceptos básicos de sensores, actuadores y programación de condiciones.

2. Organización de equipos y planificación (5 minutos)

- Distribución en parejas o grupos de tres alumnos.
- Reparto de roles (programador, técnico de montaje, coordinador de equipo).

3. Montaje físico (15 minutos)

- Ensamblaje básico de los componentes necesarios.
- Conexión del sensor de distancia a la placa del robot Maqueen.

4. Inicio de la programación (15 minutos)

- Programar la detección de presencia mediante condiciones lógicas simples (por ejemplo: si la distancia es menor a un umbral, activar la alarma).
- Pruebas iniciales del sensor y ajustes básicos.

- **Sesión 2 (45 minutos)**

Objetivo: Finalizar la programación, realizar pruebas completas y preparar la demostración final.

1. Finalización de la programación (20 minutos)

- Mejorar la lógica del sistema: ajustar la sensibilidad, añadir funciones complementarias como señales luminosas o sonoras intermitentes.
- Integrar pequeños detalles de seguridad o mejora en el código.

2. Pruebas técnicas y depuración (15 minutos)

- Comprobación del sistema en diferentes situaciones: detectar una persona, evitar falsas alarmas, asegurar el correcto funcionamiento del buzzer/LED.

3. Demostración y cierre (10 minutos)

- Cada grupo presenta brevemente el funcionamiento de su sistema.
- Discusión final sobre los principales aprendizajes y dificultades encontradas.

3.5. Atención a la Diversidad

- Trabajo en parejas para facilitar el apoyo mutuo
- Diferentes niveles de complejidad en la personalización del proyecto
- Código base predefinido para estudiantes con menos experiencia en programación
- Apoyo visual mediante diagramas de conexión claramente etiquetados
- Extensiones opcionales para alumnado avanzado

4. Ejecución y evaluación

4.1. Resultado Final

Un sistema de alarma montado y programado que:

- Detecte objetos a menos de cierta distancia.
- Active una alarma (sonido/luz).
- Esté correctamente ensamblado y documentado.

4.2. Evaluación

Rúbrica de evaluación

Aspecto	Excelente (3)	Bueno (2)	Mejorable (1)
Montaje del sistema	Conexiones precisas y ordenadas	Conexiones correctas pero algo desorganizadas	Conexiones inseguras o incorrectas
Programación	Código eficiente, bien estructurado y funcionando perfectamente	Código funcional con estructura adecuada	Código básico con errores o ineficiencias
Detección de presencia	Detección precisa y consistente a la distancia establecida	Detección correcta con algunos fallos ocasionales	Detección inconsistente o poco fiable
Sistema de alerta	Alertas visuales y sonoras claras y efectivas	Alertas funcionales pero mejorables	Alertas poco efectivas o mal implementadas
Personalización	Incorpora mejoras o personalizaciones creativas	Realiza algunas personalizaciones básicas	Implementa solo lo esencial sin personalización
Comprensión conceptual	Explica con claridad el funcionamiento y principios del sistema	Comprende los conceptos principales	Comprensión limitada del funcionamiento

Instrumentos de evaluación

- Observación directa durante el montaje y programación
- Prueba de funcionamiento del sistema final
- Explicación oral del proyecto por parte de los estudiantes
- Revisión del código programado

4.3. Recomendaciones y Ampliaciones

- Cambiar el patrón de parpadeo de los LEDs
- Modificar los tonos o melodía de la alarma
- Ajustar el umbral de detección según necesidades
- Añadir un botón para activar/desactivar la alarma
- Implementar diferentes niveles de alerta según la distancia
- Añadir un temporizador para que la alarma se apague después de un tiempo
- Incorporar un sistema de contraseña para desactivar la alarma
- Añadir un registro de detecciones (contador)

4.4. Conexión con los ODS

- **ODS 4:** Educación de calidad.
- **ODS 9:** Industria, innovación e infraestructura.
- **ODS 11:** Ciudades seguras e inclusivas.

5. Información Legal

- **Términos de uso**

Este recurso está bajo licencia Creative Commons BY-NC-SA, que permite:

- Compartir y adaptar el material
- Uso no comercial
- Compartir bajo la misma licencia
- Atribución al autor original

- **Atribuciones**

- Material gráfico:
 - www.canva.com
 - www.freepik.es
- Voces e imagen: Estefanía de Castro
- Programación: Manuel Valencia
- Contenido educativo: Manuel Valencia

ANEXO I

Mi Diario de Aprendizaje

Nombre: _____

Actividad: _____

Paso 1: Reflexiona sobre la actividad

1. ¿Qué te ha parecido la actividad? (Rodea la respuesta más adecuada)

Muy interesante Interesante Poco interesante Nada interesante

2. ¿Qué parte de la actividad te ha resultado más fácil?

3. ¿Qué parte te ha parecido más difícil?

4. ¿Cómo te has sentido mientras realizabas la actividad? (Rodea la respuesta más adecuada)

Muy bien Bien Normal Poco motivado

Paso 2: Trabajo en equipo

5. ¿Cómo ha sido tu relación con el equipo? (Marca con una X)

- Nos hemos organizado bien y hemos trabajado en equipo
- A veces hemos tenido dificultades para coordinarnos
- No hemos trabajado bien juntos

6. ¿Cómo has contribuido al equipo? (Escribe una o varias acciones realizadas)

7. ¿Qué has aprendido de trabajar con tus compañeros?

Paso 3: Evaluación y mejoras

8. ¿Qué mejorarías de la actividad para que fuera más interesante o divertida?

9. ¿Te gustaría hacer más actividades con micro:bit, Maqueen y HuskyLens?

- Sí, me ha gustado mucho
- Quizás, si fueran diferentes
- No, prefiero otro tipo de actividades

10. Escribe una palabra o frase que resuma cómo ha sido la experiencia para ti:
