


Buscando la salida

| | | | |
|---|--------------------|---|---|
| Etapas: Educación Secundaria Obligatoria | | Curso: 2º-3º | Dificultad: alta |
| Temporalización: 90 minutos | Sesiones: 2 | Áreas: Tecnología y Digitalización / Matemáticas | |
| Palabras clave: robótica, ultrasonidos, algoritmos de navegación, laberintos, programación, Maqueen, autonomía | | |  |

1. Introducción

Inspirado en Mario Kart, este reto te convierte en mecánico e ingeniero de un kart atrapado. Tu misión: programar un robot Maqueen para que encuentre la salida del circuito-laberinto del Castillo de Bowser.

El desafío consiste en:

- Programar tu kart-robot con sensor de ultrasonidos
- Hacer que navegue de manera autónoma por el circuito-laberinto
- Detectar paredes y obstáculos usando los sensores
- Programar decisiones lógicas para encontrar la ruta de escape

Tu robot deberá usar sus sensores para detectar las paredes y tomar decisiones inteligentes en cada cruce. A lo largo del recorrido podrás encontrar "cajas de objetos" (marcadores especiales) que sumarán puntos extra.

Este proyecto fomenta la comprensión de sensores, estructuras de control y resolución de problemas, todo ambientado en el universo de Mario Kart.

¿Podrá tu kart-robot encontrar la salida del castillo sin caer en las trampas de Bowser?

El reto es programar un robot Maqueen con sensor de ultrasonidos para que navegue de manera autónoma por un laberinto, detectando paredes y tomando decisiones con lógica algorítmica. Este proyecto fomenta la comprensión de sensores, estructuras de control, pensamiento computacional y resolución de problemas espaciales.

2. Marco curricular

2.1. Objetivos generales

- Comprender el funcionamiento de sensores y su aplicación en navegación autónoma.
- Diseñar e implementar algoritmos que resuelvan retos espaciales.
- Fomentar el pensamiento computacional y la programación de comportamientos inteligentes.
- Trabajar en equipo, desarrollando estrategias, depurando errores y evaluando resultados.

2.2. Competencias clave (LOMLOE)

- Competencia digital.
- Competencia STEM.
- Competencia matemática.
- Competencia personal, social y de aprender a aprender.

2.3. Competencias específicas, criterios de evaluación y saberes básicos

Área: Tecnología y Digitalización

| Competencias específicas | Criterios de evaluación | Saberes básicos |
|---|--|---|
| CE2. Resolver problemas tecnológicos con autonomía y creatividad. | C2.1 Diseñar soluciones eficaces e innovadoras. | Bloque A: Resolución de problemas; uso de sensores y materiales. |
| CE5. Desarrollar algoritmos y aplicar pensamiento computacional. | C5.1 Programar algoritmos que controlen el comportamiento de dispositivos. | Bloque B: Programación; estructuras de control condicional. |
| CE6. Comprender y usar sistemas de control programado. | C6.1 Montar y programar sistemas automáticos con sensores. | Bloque C: Robótica educativa; sensores de distancia y actuadores. |

Área: Matemáticas

| Competencias específicas | Criterios de evaluación | Saberes básicos |
|---|--|---|
| CE1. Resolver problemas matemáticos de la vida cotidiana. | C1.1 Modelizar trayectorias y representar soluciones espaciales. | Bloque D: Sentido algebraico; pensamiento computacional. |
| CE4. Aplicar principios del pensamiento computacional. | C4.1 Descomponer problemas espaciales en algoritmos efectivos. | Bloque F: Sentido socioafectivo; trabajo cooperativo y perseverancia. |

3. Planificación didáctica

3.1. Conocimientos Previos

- Programación por bloques en MakeCode.
- Conocimiento del funcionamiento de sensores de ultrasonidos.
- Manejo del robot Maqueen y micro:bit.
- Fundamentos de lógica condicional y bucles.
- Capacidad de trabajo en equipo y pensamiento analítico.

3.2. Metodología

La metodología principal es el **Aprendizaje Basado en Proyectos (ABP)**. A partir de un reto contextualizado, los estudiantes analizan, diseñan y prueban un sistema robótico para resolver un problema real.

Estrategias didácticas:

- Aprendizaje cooperativo con reparto de roles (programador, estratega, ingeniero de sensores, analista de pruebas).
- Indagación guiada sobre estrategias de resolución de laberintos.
- Resolución de problemas técnicos iterativos.
- Aprendizaje manipulativo y reflexivo.

3.3. Temporalización, Espacios, Materiales y Recursos

Sesiones: 2

Duración por sesión: 45 minutos

Resumen de sesiones:

- **Sesión 1:** Análisis del reto, programación del algoritmo de navegación y detección básica de obstáculos.
- **Sesión 2:** Ejecución del reto, integración de elementos del recorrido, recogida de puntos y optimización final.

Espacios:

- Aula de Tecnología o espacio amplio para montar laberintos.
- Espacio amplio para los laberintos (pueden ser varios laberintos pequeños)

Materiales:

- Robot Maqueen + micro:bit
- Sensor de ultrasonidos
- Laberintos de cartón o cartulina
- Ordenadores con MakeCode
- Fichas de trabajo y esquemas

3.4. Desarrollo Técnico por Sesiones

La actividad base consiste en un circuito-laberinto con 5 pasillos, donde el robot debe comenzar por la entrada del primer pasillo y, tras recorrerlo, acceder al siguiente laberinto, encadenando varios módulos hasta encontrar la salida final. Cada pasillo tiene una única salida que conecta con el siguiente módulo, simulando el avance por el Castillo de Bowser.

En una versión avanzada, opcional, los pasillos pueden presentar múltiples salidas (falsas o verdaderas), lo que obligará al robot a tomar decisiones más complejas basadas en la detección de obstáculos, favoreciendo el diseño de estrategias algorítmicas más sofisticadas y con elementos aleatorios o de puntuación mediante "cajas de objetos".

Sesión 1: Programación inicial y diseño del algoritmo (45 minutos)

- **Introducción al reto y estrategias (10 min):** Se contextualiza el uso real de robots en entornos desconocidos. Se explican algoritmos básicos: mano derecha/izquierda, flooding, Tremaux.
- **Análisis de sensores (10 min):** Funcionamiento del sensor de ultrasonidos. Calibración para distancias.
- **Diseño del algoritmo (10 min):** Diagrama de flujo y lógica de toma de decisiones.
- **Programación inicial (15 min):** Código para detectar obstáculos y realizar giros según condiciones.

Sesión 2: Pruebas, optimización y exposición (45 minutos)

- **Ajuste y mejora del algoritmo (15 min):** Ajustes de velocidad, temporizadores y condiciones.
- **Pruebas en laberinto (15 min):** Simulación real, identificación de errores y resolución.
- **Exhibición final (10 min):** Cada grupo muestra su robot en acción.
- **Reflexión final (5 min):** Registro en diario de aprendizaje y autoevaluación del proceso.

3.5. Atención a la Diversidad

- Diferentes niveles de complejidad en los laberintos según capacidades
- Posibilidad de utilizar algoritmos más sencillos para algunos equipos
- Plantillas de código con estructuras básicas para facilitar la programación
- Roles adaptados a diferentes capacidades e intereses
- Posibilidad de combinar sensores (añadir siguelíneas) para equipos que necesiten apoyo adicional

4. Ejecución y evaluación

4.1. Resultado Final

Un robot programado que:

- Navega de forma autónoma por un laberinto.
- Detecta y evita obstáculos usando ultrasonidos.
- Implementa un algoritmo de resolución de laberintos.
- Optimiza su recorrido según pruebas.

4.2. Evaluación

Rúbrica de evaluación

| Aspecto | Excelente (3) | Bueno (2) | Mejorable (1) |
|--------------------------------|---|---|--|
| Implementación del algoritmo | El robot implementa correctamente un algoritmo completo de resolución de laberintos | El robot implementa un algoritmo básico pero con algunas limitaciones | El algoritmo muestra deficiencias importantes o es muy básico |
| Uso del sensor de ultrasonidos | Utilización óptima del sensor para detectar obstáculos en diferentes direcciones | Uso correcto del sensor con algunas imprecisiones | Uso básico o impreciso del sensor |
| Navegación autónoma | El robot navega completamente autónomo y encuentra la salida eficientemente | El robot navega con cierta autonomía pero ocasionalmente se bloquea | El robot necesita intervención frecuente o no logra completar el laberinto |
| Optimización del código | Código bien estructurado, optimizado y con manejo de excepciones | Código funcional con estructura adecuada | Código básico con estructura mejorable |
| Trabajo en equipo | Excelente colaboración, con roles bien definidos y participación equilibrada | Buena colaboración con algunos desajustes | Colaboración limitada o desequilibrada |

Instrumentos de evaluación

- Observación directa del funcionamiento del robot en el laberinto
- Análisis del código programado
- Documentación del proceso (diagramas de flujo, estrategias)
- Reflexión final sobre el proceso y los resultados
- Coevaluación entre equipos

4.3. Recomendaciones y Ampliaciones

- **Competición de laberintos:** Organizar una competición para ver qué robot resuelve el laberinto más rápido
- **Laberintos con variaciones:** Incluir elementos móviles o zonas con diferentes texturas
- **Mapeo del laberinto:** Programar al robot para que construya un mapa del laberinto mientras lo explora
- **Combinación de sensores:** Añadir sensores siguelíneas o de color para enriquecer la navegación
- **Sensor siguelíneas automatizado:** Añadir un servomotor sobre el que se monta el sensor de distancia y detecta el laberinto moviendo solo el servo.
- **Inteligencia colectiva:** Hacer que varios robots colaboren en la exploración del laberinto

Extensión con HuskyLens (opcional)

Si se dispone de la cámara HuskyLens, los alumnos podrían programarla para reconocer marcas visuales que indiquen caminos preferentes o la salida, combinando navegación por ultrasonidos con reconocimiento visual.

4.4. Conexión con los ODS

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

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: Manuel Valencia
 - 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:
