

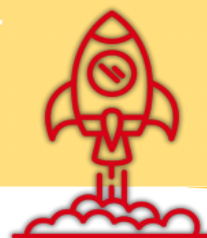


Título: π -Robot: El Escáner de circunferencias

Nivel educativo: 6º Educación Primaria

Áreas Curriculares: Matemáticas, Tecnología y Robótica

Temporalización: 14 de marzo (día de π)



Descripción breve de la actividad

El alumnado diseñará y construirá su propio π -Robot, un robot con sensor de ultrasonidos que actuará como escáner inteligente para comprobar cómo las matemáticas explican la realidad.

El reto tiene dos partes:

1. Cero magia, mucha lógica

El robot medirá el radio de objetos circulares del aula y, mediante el programa creado por el alumnado, calculará la circunferencia. Al obtener un valor cercano a 3,14 comprobarán que π surge de la medición real, no del libro.

2. ¿Somos proporcionales?

Con el mismo sensor, el robot medirá partes del cuerpo para analizar su aproximación a la proporción áurea ($\phi \approx 1,618$), contrastando los resultados con el modelo del *Hombre de Vitruvio*.

La actividad integra sensores, variables y operadores aplicados a un enigma matemático y artístico de hace más de 500 años, fomentando el pensamiento computacional desde la experimentación.





Objetivos

- Identificar relaciones métricas en figuras circulares (relación diámetro/longitud).
- Programar un objeto interactivo para recoger datos del entorno mediante sensores.
- Utilizar el pensamiento computacional para automatizar cálculos matemáticos.

Competencias clave a desarrollar: Competencia digital y Competencia lingüística.



¿Cómo lo hacemos?

FASE 1: Investigación Teórica (El Ojo del Genio)

Objetivo: Comprender la geometría aplicada al arte y la naturaleza.

- **Análisis del "Hombre de Vitruvio":** El alumnado realizará un análisis crítico de la obra para identificar el ombligo como el centro geométrico de la circunferencia que circunscribe la figura.
- **Proporciones y Constantes:** Se investigará la tendencia de las proporciones humanas hacia el **Número Áureo** ($\phi \approx 1,618$). Asimismo, se realizará una toma de datos manual midiendo objetos circulares para introducir la constante π ($\approx 3,1416$) mediante la relación entre perímetro y diámetro.

FASE 2: Construcción y Prototipado

Objetivo: Ensamblar el hardware del robot móvil utilizando el chasis MiniQ de DFRobot.

- **Montaje Mecánico:** Por equipos, se ensamblarán los motores de corriente continua (CC), las ruedas y la rueda loca (bola de apoyo) sobre el chasis.



- **Electrónica y Estructura:** Se instalará la placa de expansión para micro:bit y el portapilas, asegurando que los cables queden recogidos y no obstruyan el movimiento de los motores. Al insertar la micro:bit, se debe verificar que la pantalla LED quede orientada hacia el exterior.
- **Caracterización:** Se acoplará la plantilla de Leonardo da Vinci al chasis y se fijará el **sensor de ultrasonidos** en la parte frontal, asegurando que esté bien alineado para una medición óptima.

FASE 3: Programación y Toma de Datos

Objetivo: Desarrollar el algoritmo de medición en el entorno MakeCode.

- **Configuración:** En makecode.microbit.org, se instalará la extensión específica de DFRobot para el kit de Robótica Creativa.
- **Programación:** Se desarrollará el código para que el robot avance, detecte el obstáculo y aplique la fórmula matemática de la circunferencia.
- **Registro:** Una vez ejecutado el programa, el alumnado anotará los resultados calculados por la micro:bit en su cuaderno de trabajo.

FASE 4: Validación y Depuración (Debugging)

Objetivo: Contrastar el modelo digital con la medición física real.

- **Comprobación:** Se verificará el cálculo del robot utilizando una cinta métrica.
- **Análisis de Errores:** En caso de discrepancia, se analizará el rango de error para identificar si el fallo es de programación (lógica) o de ensamblaje (mecánica/sensores).
- **Optimización:** Se aprovechará esta fase para ajustar la velocidad de los motores y corregir cualquier imprecisión detectada en el comportamiento del robot.
- si el error es de **hardware** (¿el sensor detecta el suelo en lugar del objeto?) o de **software** (¿hemos redondeado demasiado el número Pi?).
- **Ajuste de Motores:** Si el robot se desvía al avanzar, los alumnos deben ajustar ligeramente la potencia de un motor frente al otro en el código para equilibrar la trayectoria.



Sugerencias

En el caso de que a algún alumno tenga algún tipo de dificultad con el movimiento del robot, se puede hacer la misma actividad adaptada, eliminando la parte de la programación en la que el robot se mueve.

Para alumnos que resuelvan con facilidad este reto, se propone añadir un aviso de proximidad a medida que se aproxime a la pared, como si de un sensor de parking se tratara.

Recursos

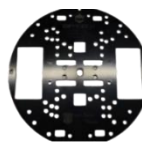
- **Personales:** Equipo docente de 6º y alumnado.
- **Materiales:** Placa micro:bit (V2), Placa de expansión IO (MBT0014), Chasis MiniQ y Sensor ultrasónico (SEN0388).
 - Plantilla Leonardo Robot.
 - Ordenador con acceso a internet por equipos.



Motor CC
y ruedas



Sensor ultrasónicos



Chasis



Porta-pilas



Placa de Expansión y micro:bit



Espacios: Aula del Futuro o Aula de Tecnología

Tipo de actividad: Grupos de trabajo cooperativo



Vídeo presentación: [Ver vídeo](#)

Plataforma de programación: makecode.microbit.org

Materiales para descargar:

[Plantilla de Leonardo](#)

[Programación .hex](#)



Imagen creada con IA.

```

al presionarse el botón A
  mientras Obtener distancia del sensor ultrasónico del pin P0 en (cm). > 20
    ejecutar Motor All dirección Cx velocidad 50
  Parar motor All
  repetir 4 veces
    ejecutar mostrar cadena C
    mostrar número circunferencia
  mostrar ícono

al iniciar
  mostrar LEDs
  fijar radio a Obtener distancia del sensor ultrasónico del pin P0 en (cm).
  fijar circunferencia a π x 2 x radio

al presionarse el botón B
  fijar altura_total a Obtener distancia del sensor ultrasónico del pin P0 en (cm).

al presionarse el botón A+B
  fijar distancia_ombligo a Obtener distancia del sensor ultrasónico del pin P0 en (cm).
  fijar resultado a altura_total / distancia_ombligo
  si resultado > 1.5 y resultado < 1.7 entonces
    mostrar ícono
  si no
    mostrar ícono
  
```





¿Qué hemos aprendido?



Criterios de Evaluación	4 Excelente	3 Satisfactorio	2 Mejorable	1 Insuficiente
Algoritmos y Variables	Programa con variables y cálculos correctos con autonomía.	Programa con variables y pocos errores.	Usa variables con ayuda y errores de cálculo.	No usa variables ni aplica los cálculos.
Patrones y proporciones	Explica y relaciona π y ϕ con las mediciones.	Identifica π y ϕ en la actividad.	Reconoce los conceptos con ayuda.	No identifica π ni ϕ .
Montaje y hardware	Montaje correcto y estable, resuelve pequeños problemas.	Montaje correcto con poca ayuda.	Montaje incompleto o con ayuda frecuente.	No completa el montaje.
Medición y validación	Compara datos y explica posibles errores.	Compara mediciones básicas.	Mide pero no analiza resultados.	No realiza comprobaciones.



Pensamiento computacional

Algoritmos (pasos y reglas): seguir una serie de pasos o instrucciones bien definidas para resolver un problema o completar una tarea.

Patrones (detectar y usar similitudes): identificar similitudes o patrones en problemas o datos, lo que facilita encontrar soluciones más rápidas y eficientes.

Abstracción (eliminar detalles innecesarios): Simplificar un problema eliminando detalles que no son importantes, para enfocarse en lo que es relevante y esencial.



Más información

Códigos QR vinculados con los recursos de la actividad:



Vídeo Presentación



Makecode



Plantilla Leonardo



Archivo .hex

Autoría

Esta actividad ha sido realizada por **María del Carmen Mancheño Sáez**, en el marco del **Programa Código Escuela 4.0 _Madrid**.