



## Bluetooth riders: conduciendo un Maqueen

**Ciclo:** primer ciclo

**Curso:** 2º E.S.O.

**Áreas Curriculares:** Tecnología y digitalización

**Temporalización:** 3<sup>er</sup> trimestre

**Nº de sesiones:** 6 sesiones de 55 minutos



### Descripción del proyecto

En este proyecto los alumnos van a diseñar una aplicación móvil para controlar de forma remota el robot Mcqueen, utilizando el módulo Bluetooth que lleva integrado la placa Micro:Bit. Para la programación del robot y el dispositivo controlador, se va a trabajar de forma paralela en el entorno MakeCode y en el entorno APP Inventor.





## Objetivos

1. **Comprender el funcionamiento básico de los sistemas de control y comunicación digital**, identificando los elementos de hardware y software que intervienen en la interacción entre dispositivos (Micro:Bit, módulo Bluetooth, smartphone, robot McQueen).
2. **Desarrollar algoritmos y programas sencillos** en entornos visuales (MakeCode y App Inventor) para resolver problemas de control y comunicación entre dispositivos.
3. **Diseñar y construir aplicaciones digitales funcionales**, aplicando principios básicos de diseño de interfaces y usabilidad.
4. **Explorar el uso de sensores, actuadores y módulos de comunicación** para crear sistemas interactivos que respondan a órdenes o estímulos.
5. **Trabajar de forma cooperativa en proyectos tecnológicos**, asumiendo roles y responsabilidades dentro del grupo, respetando las ideas ajenas y contribuyendo al desarrollo común.
6. **Aplicar el método de proyectos en el diseño y desarrollo de soluciones tecnológicas**, planificando tareas, gestionando el tiempo y documentando el proceso.
7. **Fomentar la creatividad y el pensamiento computacional, resolviendo retos mediante la descomposición de problemas**, la abstracción y la depuración de errores.
8. **Valorar el impacto de la tecnología y la digitalización en la sociedad actual**, reflexionando sobre su uso responsable, ético y sostenible.
9. **Integrar conocimientos de distintas áreas STEAM** (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas) para la creación de un producto tecnológico funcional.

## COMPETENCIAS ESPECÍFICAS

**CE1.** Identificar y comprender los elementos y sistemas tecnológicos presentes en su entorno.

**CE2.** Aplicar estrategias de diseño y pensamiento computacional para resolver problemas tecnológicos.



**CE3.** Desarrollar productos o soluciones digitales de forma creativa y colaborativa.

**CE4.** Utilizar de manera responsable las tecnologías digitales, considerando aspectos éticos y de seguridad.

**CE5.** Comunicar y documentar procesos y resultados de proyectos tecnológicos.



## Contenidos

### 1. Sistemas digitales y comunicaciones:

- Analizar y dibujar el esquema funcional del sistema (smartphone → Bluetooth → Micro:Bit → robot).
- Identificar módulos físicos (placa, módulo Bluetooth) y su conexión física.
- Configurar parámetros de comunicación (dirección, emparejamiento).

### 2. Programación y lógica de control:

- Diseñar algoritmos para controlar movimientos (adelante, atrás, derecha, izquierda).
- Programar en MakeCode el Micro:Bit para recibir datos Bluetooth y traducirlos en acciones del robot.
- Crear lógica de control en la app (App Inventor) para generación de comandos (pulsadores, joystick virtual).
- Probar, depurar y corregir errores del sistema.
- Simular trayectorias, detener ante obstáculos (si se añade sensor).

### 3. Robótica, sensores y actuadores:

- Conectar motor(es) del robot al Micro:Bit, siguiendo esquemas.
- Integrar sensores básicos (opcional) para decisiones automáticas o de seguridad.
- Leer valores de sensores y utilizarlos en condicionales (por ejemplo, detener si obstáculo).

### 4. Diseño y creación de aplicaciones digitales:

- Diseñar bocetos de interfaz (wireframes) para la app de control.
- Crear la app en App Inventor con botones o controles que envíen comandos Bluetooth.
- Procesar datos recibidos (feedback del robot) en la app.
- Probar y ajustar la comunicación entre app y hardware.
- Versionar y mejorar la interfaz con feedback del grupo o usuarios.

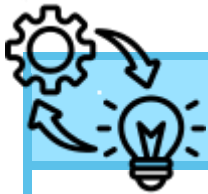


## 5. Proyecto tecnológico integrado / sistema embebido:

- Planificar el cronograma del proyecto (tareas, hitos).
- Coordinar en grupo: quién hace qué (hardware, programación, interfaz, pruebas).
- Integrar todos los componentes (robot + app) en un sistema operativo.
- Realizar pruebas finales completas.
- Documentar el proyecto: esquema, código, capturas de pantalla, informe, presentación.

## 6. Ética, seguridad y digitalización responsable:

- Configurar la app/robot para evitar accesos no deseados (por ejemplo, un código de emparejamiento, rechazo de comandos desconocidos).
- Evaluar el consumo energético del sistema y buscar mejoras (apagar módulos cuando no estén en uso).
- Reflexionar y debatir en clase sobre usos positivos y riesgos de robots controlados por app.



## Metodología

En el desarrollo de este proyecto, se han seleccionado diversas metodologías didácticas. Estas han sido elegidas por su capacidad para involucrar activamente a los estudiantes y promover un aprendizaje significativo.

A continuación, se describen los enfoques metodológicos que se implementarán:

### - Aprender haciendo (Learning by doing):

Esta metodología, permite que el alumnado aprenda mediante la creación de artefactos funcionales como apps, interfaces y robots. A través de la manipulación directa de componentes (micro:bit, Maqueen, App Inventor), los estudiantes desarrollan habilidades técnicas y comprenden mejor los conceptos de programación, electrónica y diseño digital.

### - Aprendizaje Basado en Retos (ABR):

El proyecto se estructura en torno a un reto concreto: diseñar y programar una app que controle un robot mediante Bluetooth. Este enfoque promueve el pensamiento crítico, la resolución de problemas y la toma de decisiones, ya que los estudiantes deben superar obstáculos técnicos y creativos para lograr un producto funcional.



- **Tutoría entre iguales:**  
Los grupos se conforman de manera heterogénea, permitiendo que los estudiantes con diferentes niveles de competencia colaboren y aprendan unos de otros. Esta tutoría entre iguales favorece la inclusión, la empatía y el desarrollo de habilidades sociales, permitiendo que todos los alumnos participen activamente y se sientan parte del proceso.
- **Aprendizaje cooperativo:**  
Cada equipo de trabajo asume roles definidos (programador, diseñador, documentador, probador, portavoz), fomentando la interdependencia positiva y la responsabilidad individual. Esta estructura permite que todos los miembros contribuyan al éxito del grupo, desarrollando habilidades de comunicación, planificación y gestión del tiempo.
- **Aprendizaje Basado en Proyectos (ABP):**  
El alumnado trabaja en un proyecto completo que integra conocimientos de tecnología, matemáticas, diseño y comunicación. A lo largo de varias sesiones, planifican, diseñan, programan, prueban y presentan su solución, lo que les permite profundizar en los contenidos y desarrollar competencias clave de forma transversal.
- **Flipped Classroom (Aula Invertida):**  
Se propone el uso del aula virtual (Educamadrid) para que el alumnado acceda a videotutoriales, guías visuales y ejemplos de código antes de las sesiones prácticas. Esto permite aprovechar el tiempo de clase para resolver dudas, realizar pruebas y avanzar en el proyecto con el acompañamiento del docente.
- **Aprendizaje Basado en Problemas:**  
Durante el desarrollo del proyecto, los estudiantes se enfrentan a problemas reales como errores de conexión Bluetooth, fallos en la lógica de programación o ajustes de velocidad del robot. Resolver estos problemas les obliga a investigar, experimentar y aplicar sus conocimientos de forma autónoma y significativa.



## Preguntas para la gestión del aula

Para gestionar día a día mi trabajo, me planteo las siguientes preguntas:

### **¿Cómo organizo los grupos para que todos participen en el diseño y control del robot?**

Con 30 alumnos y 6 mesas de trabajo, lo ideal es formar grupos de 5 estudiantes con perfiles variados. Cada grupo puede asumir roles rotativos (programador, diseñador de interfaz, probador, documentador, portavoz) para que todos participen activamente en las distintas fases del proyecto. Esta rotación permite que cada alumno experimente diferentes responsabilidades y desarrolle habilidades técnicas, comunicativas y organizativas.

### **¿Cuándo es el mejor momento para repartir los kits de robótica y los dispositivos?**

Dependerá del tipo de sesión. Si se va a introducir un nuevo concepto (como la conexión Bluetooth o la estructura de bloques en App Inventor), es preferible esperar a repartir el material para asegurar la atención. En cambio, si la sesión es práctica desde el inicio (como pruebas de movimientos o personalización de la app), se puede entregar el material por fases, según el avance de cada grupo. Esto permite mantener el foco y evitar distracciones.

### **¿Qué hago si un grupo no consigue conectar la app con la micro:bit por Bluetooth?**

Es una situación habitual. Puedes preparar una guía visual con los pasos de emparejamiento, revisar permisos de Android y comprobar que la micro:bit esté encendida y visible. Si el problema persiste, puedes proponer que simulen los movimientos con la app sin conexión mientras otro grupo termina y libera un kit funcional. Así no se detiene el aprendizaje y se mantiene la motivación.

### **¿Cómo gestiono el tiempo cuando algunos grupos terminan antes que otros?**

Puedes ofrecer retos opcionales de ampliación, como diseñar nuevos botones con funciones combinadas (giro + avance), personalizar la interfaz con iconos o programar respuestas más complejas. También puedes proponer que documenten su proceso en una ficha visual o preparen una presentación para compartir con el resto de la clase. Esto mantiene el ritmo activo y da valor al tiempo extra.



## ¿Qué hago si un grupo se bloquea y depende demasiado de mí para avanzar?

Puedes derivarlos al aula virtual, donde encontrarán videotutoriales, ejemplos de bloques y guías paso a paso. También puedes plantearles preguntas orientadoras o dividir la tarea en pasos más pequeños. El objetivo es que recuperen la confianza y avancen de forma autónoma, sin necesidad de intervención constante.

## ¿Cómo puedo aprovechar el aula virtual de Educamadrid durante este proyecto?

El aula virtual puede contener los objetivos del proyecto, videotutoriales sobre App Inventor y MakeCode, ejemplos de código, rúbricas de evaluación y fichas de reflexión. También puedes habilitar la entrega de tareas (como capturas de pantalla de la app o vídeos de las pruebas del robot) para recoger evidencias del trabajo. Esto refuerza la autonomía y permite que el aprendizaje continúe fuera del aula.

## ¿Cómo organizo los materiales si no se termina el proyecto en una sola sesión?

Puedes asignar a cada grupo una caja etiquetada donde guarden su micro:bit, robot, cables y fichas. Estas cajas pueden almacenarse en estanterías por curso o grupo. Al final de cada sesión, los alumnos deben recoger y guardar todo su material, y tú puedes supervisar antes de salir del aula. Esto garantiza continuidad, orden y responsabilidad compartida.

## ¿Qué hago si no tengo suficientes kits para todos los grupos a la vez?

Puedes organizar el proyecto en fases: mientras unos grupos trabajan con el robot, otros pueden diseñar la interfaz en App Inventor, documentar el proceso o preparar la presentación. También puedes alternar sesiones en el aula de informática para trabajar el diseño digital o la edición de vídeos. Así se optimiza el uso del material y se mantiene el ritmo del proyecto.

## ¿Qué recursos visuales pueden ayudar a la dinámica del aula?

Puedes usar paneles o corchos con esquemas de conexión, ejemplos de bloques de código, plantillas de interfaz o rúbricas de evaluación. También puedes dejar a la vista un “muro de soluciones” con errores comunes y cómo resolverlos. Estos apoyos visuales permiten que los alumnos consulten de forma autónoma y reduzcan la dependencia del docente.

## ¿Cómo evalúo individualmente si el trabajo es en grupo?

Asignando roles rotativos y tareas específicas a cada alumno, puedes observar su



implicación y recoger evidencias individuales. Además, puedes aplicar una rúbrica de autoevaluación y coevaluación, y complementar con una prueba objetiva o una reflexión escrita. Así garantizas una evaluación justa y formativa, tanto del proceso como del producto final.



## Recursos

Personales	Materiales	Digitales
<p><b>Docentes:</b> Profesores que guían el proceso, explican los conceptos técnicos, facilitan el uso de App Inventor y MakeCode, y acompañan en la resolución de problemas durante el diseño y la programación del robot.</p> <p><b>Estudiantes:</b> Compañeros que colaboran en grupos, comparten ideas, se reparten tareas y aprenden juntos a través de la práctica y la observación entre iguales.</p>	<p><b>Cuadernos y fichas impresas:</b> Se utilizan para registrar ideas, anotar comandos, dibujar esquemas de interfaz, documentar pruebas del robot y reflexionar sobre el proceso. También se incluyen rúbricas de evaluación y plantillas de observación entre equipos.</p> <p><b>Material didáctico específico:</b></p> <ul style="list-style-type: none"><li>- Fichas visuales con ejemplos de bloques de MakeCode y App Inventor.</li><li>- Esquemas de conexión entre micro:bit y Maqueen.</li><li>- Plantillas para diseñar la app en</li></ul>	<p><b>Plataformas educativas:</b></p> <ul style="list-style-type: none"><li>- Aula Virtual de Educamadrid como espacio central para:</li><li>- Publicar los objetivos del proyecto.</li><li>- Compartir videotutoriales y guías paso a paso.</li><li>- Subir evidencias del trabajo (capturas de pantalla, vídeos, reflexiones).</li><li>- Acceder a rúbricas y listas de cotejo.</li></ul> <p><b>Aplicaciones y software educativo:</b></p> <ul style="list-style-type: none"><li>- <b>App Inventor:</b> para diseñar la interfaz de control</li></ul>



# Proyecto didáctico



	<p>papel antes de programarla.</p> <ul style="list-style-type: none"><li>- Rúbricas de autoevaluación y coevaluación.</li><li>- Circuitos de prueba (laberintos, recorridos con cinta adhesiva).</li></ul> <p><b>Tecnología:</b></p> <ul style="list-style-type: none"><li>- Micro:bit con alimentación por batería o cable USB.</li><li>- Robot Maqueen compatible con micro:bit.</li><li>- Dispositivos móviles o tabletas con App Inventor instalado.</li><li>- Ordenadores con acceso a MakeCode.</li><li>- Proyector o pizarra digital para mostrar ejemplos y bloques en tiempo real.</li></ul> <p><b>Material de aula:</b></p> <ul style="list-style-type: none"><li>- Cajas etiquetadas para guardar los kits por grupo.</li><li>- Cinta adhesiva, cartulinas,</li></ul>	<p>del robot y programar el envío de comandos por Bluetooth.</p> <ul style="list-style-type: none"><li>- <b>MakeCode:</b> para programar la micro:bit y definir las respuestas del robot a los comandos recibidos.</li><li>- <b>Emuladores:</b> para probar la app o el código sin necesidad de conexión física, especialmente útil si no hay kits suficientes.</li></ul> <p><b>Recursos en línea:</b></p> <ul style="list-style-type: none"><li>- Vídeos explicativos sobre cómo emparejar la micro:bit por Bluetooth, cómo usar bloques condicionales o cómo diseñar una interfaz funcional.</li><li>- Artículos y blogs educativos con ejemplos de proyectos similares.</li><li>- Foros o comunidades educativas donde compartir dudas o soluciones (como</li></ul>
--	--	--



# Proyecto didáctico

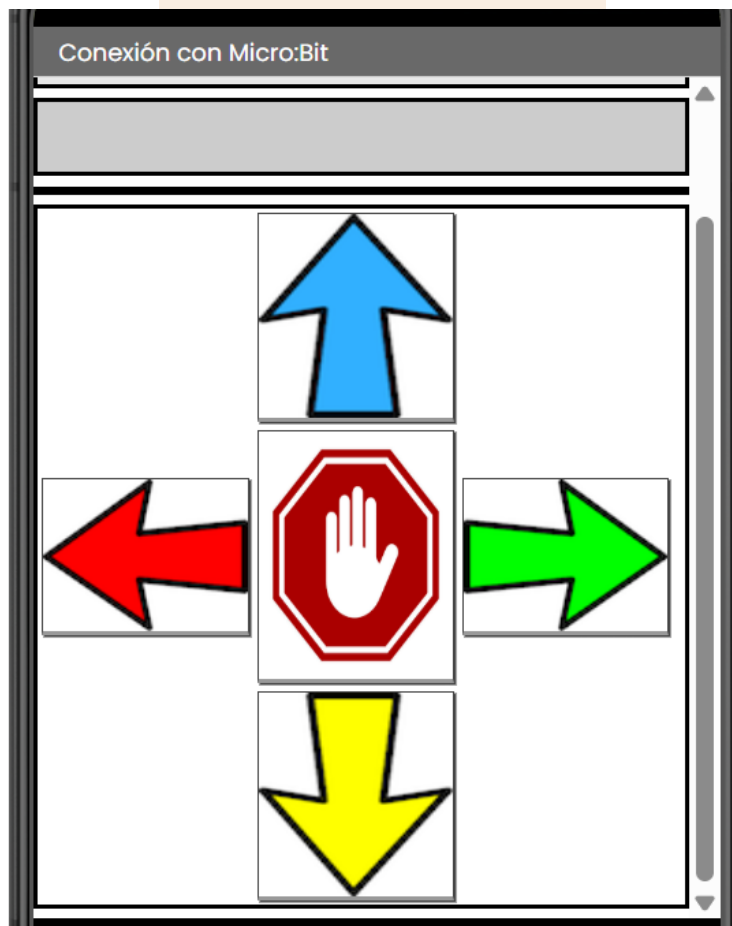
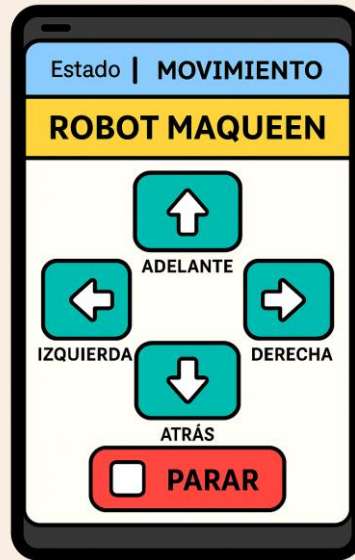


	<p>rotuladores y etiquetas para diseñar circuitos o señalar recorridos.</p> <ul style="list-style-type: none"><li>- Herramientas básicas (destornilladores, pinzas) si se requiere ajustar algún componente.</li></ul>	<p>micro:bit España o foros de App Inventor).</p>
--	--	---



## Actividades

<b>Fase</b>	1
<b>Temporalización</b>	55 minutos
<b>Tipo de Actividad</b>	Exploratoria y grupal
<b>Descripción</b>	<p>En primer lugar, el alumnado trabajará en la <b>creación de una interfaz básica en App Inventor</b> que permita controlar el robot Maqueen mediante botones táctiles. Para ello, se les propondrá diseñar una aplicación sencilla con cuatro botones principales: avanzar, retroceder, girar a la izquierda y girar a la derecha. Opcionalmente, podrán añadir un botón de parada y etiquetas que indiquen el estado del robot. Se fomentará la personalización visual de la interfaz, utilizando colores contrastados, iconos intuitivos y una disposición clara que facilite la interacción. Esta actividad permite desarrollar habilidades de diseño digital y pensamiento lógico, además de introducir la estructura de una app funcional.</p>



A continuación, se abordará la **programación de comandos**



en **MakeCode** para micro:bit, sin necesidad de conexión Bluetooth. El alumnado programará la micro:bit para que, al presionar sus botones físicos, el robot Maqueen ejecute movimientos básicos. Por ejemplo, el botón A hará que el robot avance, el botón B lo hará retroceder, y la combinación A+B lo detendrá. También se podrán utilizar gestos como agitar o inclinar la micro:bit para activar giros. Esta programación se realizará mediante bloques visuales en la plataforma MakeCode, lo que facilita la comprensión de la lógica secuencial y el control de eventos.

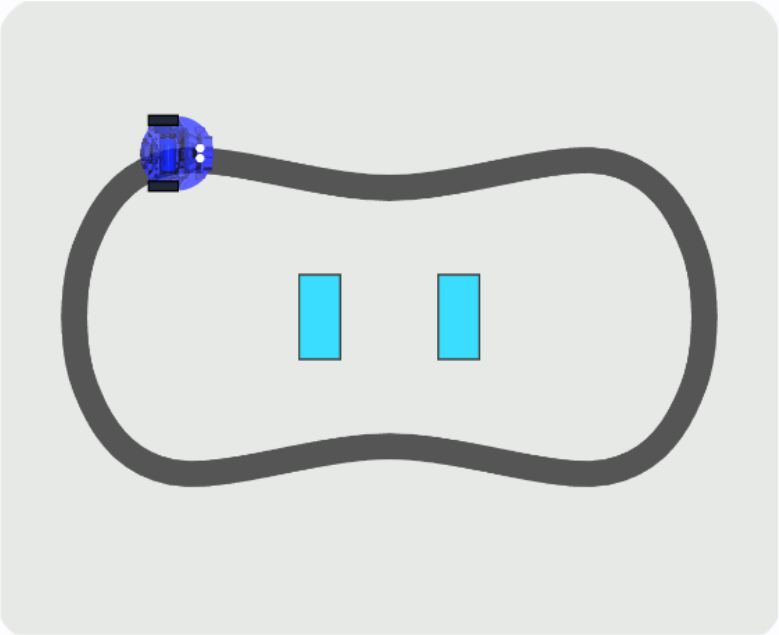
```
al iniciar
si message = "CASO 1" entonces
  Parar motor ambos
  Motor ambos sentido avanzar velocidad velocidad
si no, si message = "CASO 2" entonces
  Parar motor ambos
  Motor ambos sentido retroceder velocidad velocidad
si no, si message = "CASO 3" entonces
  Parar motor ambos
  Motor derecho sentido avanzar velocidad velocidad
  Motor izquierdo sentido retroceder velocidad velocidad
si no, si message = "CASO 4" entonces
  Parar motor ambos
  Motor derecho sentido retroceder velocidad velocidad
  Motor izquierdo sentido avanzar velocidad velocidad
si no, si message = "CASO 5" entonces
  Parar motor ambos
```

Se animará a los estudiantes a probar sus programas en el **simulador** antes de descargarlos en la micro:bit.



# Proyecto didáctico

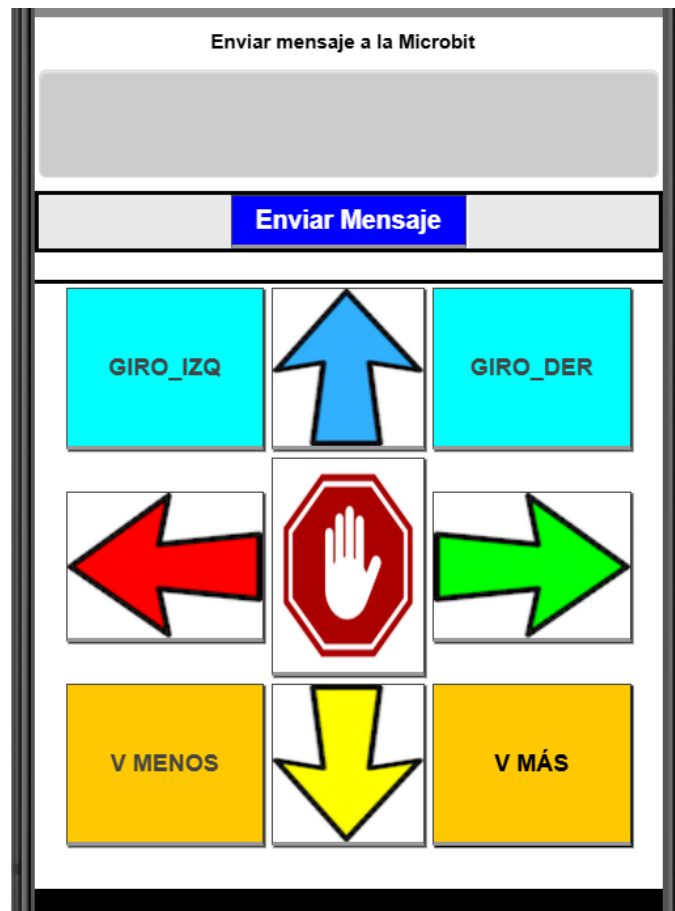


	 <p>Finalmente, se realizan <b>pruebas de funcionamiento</b> con el robot.</p>
<b>Recursos</b>	<ul style="list-style-type: none"> <li>- App Inventor v2.</li> <li>- MakeCode.</li> <li>- Robot Maqueen.</li> </ul>

<b>Fase</b>	<b>2</b>
<b>Temporalización</b>	55 minutos
<b>Tipo de Actividad</b>	Práctica guiada
<b>Descripción</b>	<p>Una vez creada la interfaz básica en App Inventor, se propone al alumnado realizar una <b>personalización visual de la aplicación</b>, adaptando colores, iconos y nombres de los botones para mejorar la usabilidad y el atractivo. Esta fase permite trabajar el diseño digital desde una perspectiva funcional y estética. Se anima a los estudiantes a elegir colores que faciliten la identificación de cada acción (por ejemplo, verde para avanzar, rojo para detener), incorporar iconos representativos (flechas, símbolos de giro) y renombrar los botones con términos claros o creativos. Esta</p>



actividad fomenta la expresión visual, la accesibilidad y la conexión emocional con el producto creado, además de reforzar la comprensión de la interfaz como herramienta de comunicación.



En paralelo, se desarrollará el **diseño de movimientos combinados en MakeCode**, utilizando la micro:bit para controlar el robot Maqueen sin conexión Bluetooth. El alumnado programará secuencias que integren varias acciones encadenadas, como girar y avanzar, retroceder y detener, o realizar un giro en U. Para ello, se utilizarán bloques de control de motores con pausas temporizadas, lo que permite modular la duración y la intensidad de cada movimiento. Esta fase introduce la lógica secuencial y el pensamiento algorítmico, y permite explorar variantes según el recorrido o el reto planteado. Se anima a los equipos a probar distintas combinaciones y ajustar parámetros como velocidad, tiempo de giro o dirección, favoreciendo la experimentación y la mejora iterativa.



```
al iniciar
si message = "CASO 1" entonces
  Parar motor ambos
  Motor ambos sentido avanzar velocidad velocidad
si no, si message = "CASO 2" entonces
  Parar motor ambos
  Motor ambos sentido retroceder velocidad velocidad
si no, si message = "CASO 3" entonces
  Parar motor ambos
  Motor izquierdo sentido avanzar velocidad velocidad
  Motor derecho sentido retroceder velocidad velocidad
si no, si message = "CASO 4" entonces
  Parar motor ambos
  Motor izquierdo sentido retroceder velocidad velocidad
  Motor derecho sentido avanzar velocidad velocidad
si no, si message = "CASO 5" entonces
  Parar motor ambos
si no, si message = "CASO 6" entonces
  cambiar velocidad por 5
si no, si message = "CASO 7" entonces
  cambiar velocidad por -5
si no, si message = "CASO 8" entonces
  Motor izquierdo sentido avanzar velocidad velocidad + 5
  Motor derecho sentido avanzar velocidad velocidad
si no, si message = "CASO 9" entonces
  Motor izquierdo sentido avanzar velocidad velocidad
  Motor derecho sentido avanzar velocidad velocidad + 5
```

Finalmente, se propone la documentación del **proceso en una ficha visual**, que recoja los pasos seguidos, las decisiones tomadas y las dificultades superadas. Esta ficha puede incluir capturas de pantalla de la app, esquemas de



# Proyecto didáctico



	los bloques de programación, anotaciones sobre los movimientos diseñados y reflexiones personales. Se trata de una herramienta clave para fomentar la metacognición, facilitar la evaluación y compartir el trabajo con otros grupos o docentes. Además, permite visibilizar el proceso creativo y técnico, reforzando el valor del aprendizaje más allá del producto final.
<b>Recursos</b>	<ul style="list-style-type: none"><li>- Robots Maqueen.</li><li>- Placa Micro:bit</li><li>- Móviles/tablets.</li><li>- <a href="#">Guía visual de MakeCode programación de movimientos combinados.</a></li></ul>

<b>Fase</b>	<b>3</b>
<b>Temporalización</b>	55 minutos
<b>Tipo de Actividad</b>	Programación por bloques
<b>Descripción</b>	En esta tercera sesión, el alumnado se centrará en la <b>programación de la interfaz creada en App Inventor</b> , estableciendo la lógica que permite enviar comandos al robot Maqueen a través de la micro:bit mediante conexión Bluetooth. Cada botón de la app —como “Avanzar”, “Retroceder”, “Girar izquierda”, “Girar derecha” y “Detener”— será vinculado a un bloque de código que envíe una cadena de texto específica al puerto serie Bluetooth. Por ejemplo, al presionar el botón “Avanzar”, se enviará el texto “A”, que la micro:bit interpretará para activar los motores del robot en sentido de avance. Esta estructura se repetirá para cada acción, utilizando bloques como BluetoothClient.SendText y asegurando que la app esté correctamente emparejada con la micro:bit.



# Proyecto didáctico



```
cuando Avanzar .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    Llamar SendText
      cmd message . Texto
    poner message . Texto como " A "
  else
    llamar Notifier1 .MostrarAlerta
      aviso " Conectar primero "

cuando Retroceder .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " B "
    Llamar SendText
      cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
      aviso " Conectar primero "

cuando Derecha .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " C "
    Llamar SendText
      cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
      aviso " Conectar primero "

cuando Izquierda .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " D "
    Llamar SendText
      cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
      aviso " Conectar primero "

cuando Parar .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " E "
    Llamar SendText
      cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
      aviso " Conectar primero "
```



"Programa financiado por el Ministerio de Educación,  
Formación Profesional y Deportes"





```
cuando Vmas .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " F "
    Llamar SendText
    cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
    aviso " Conectar primero "

cuando Vmenos .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " G "
    Llamar SendText
    cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
    aviso " Conectar primero "

cuando Giro_der .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " H "
    Llamar SendText
    cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
    aviso " Conectar primero "

cuando Giro_izq .Clic
ejecutar
  si BluetoothLE1 . IsDeviceConnected
  entonces
    poner message . Texto como " I "
    Llamar SendText
    cmd message . Texto
  else
    llamar Notifier1 .MostrarAlerta
    aviso " Conectar primero "
```

En paralelo, se trabajará la **programación en MakeCode de la micro:bit**, que debe estar preparada para recibir los comandos enviados desde la app y ejecutar las acciones correspondientes en el robot. El alumnado programará la micro:bit para que, al recibir una cadena de texto por el puerto serie, se active el movimiento deseado. Por ejemplo, si recibe "A", el Maqueen avanzará; si recibe "B", retrocederá; si recibe "C" o "D", girará en la dirección indicada; y si recibe "E", se detendrá. Esta lógica se implementa con bloques como on serial received, if...then, y maqueen.motorRun. Se



fomentará el uso de estructuras condicionales claras y la organización del código para facilitar su comprensión y depuración.

```
al iniciar
si <message> = "A" entonces
  Parar motor ambos
  Motor ambos sentido avanzar velocidad velocidad
si no, si <message> = "B" entonces
  Parar motor ambos
  Motor ambos sentido retroceder velocidad velocidad
si no, si <message> = "C" entonces
  Parar motor ambos
  Motor derecho sentido avanzar velocidad velocidad
  Motor izquierdo sentido retroceder velocidad velocidad
si no, si <message> = "D" entonces
  Parar motor ambos
  Motor derecho sentido retroceder velocidad velocidad
  Motor izquierdo sentido avanzar velocidad velocidad
si no, si <message> = "E" entonces
  Parar motor ambos
```


Finalmente, se realizan **pruebas de funcionamiento con el robot**, conectando la micro:bit al Maqueen y emparejando el dispositivo móvil con la micro:bit mediante Bluetooth. El alumnado comprobará si los comandos enviados desde la app se ejecutan correctamente en el robot, observando la respuesta a cada botón y ajustando parámetros si es necesario. Se propondrán recorridos simples o retos de navegación para validar el funcionamiento, como avanzar en línea recta, girar en esquinas o detenerse en puntos concretos. Esta fase permite consolidar el aprendizaje técnico, fomentar el trabajo colaborativo y reforzar la



# Proyecto didáctico



	motivación mediante la interacción directa con el robot.
<b>Recursos</b>	<ul style="list-style-type: none"> <li>- Ordenadores/tablets.</li> <li>- App Inventor.</li> <li>- Maqueen + Micro:bit</li> <li>- <a href="#">Fichas visuales de bloques App Inventor programación botones de movimientos.</a></li> </ul>

<b>Fase</b>	<b>4</b>
<b>Temporalización</b>	55 minutos
<b>Tipo de Actividad</b>	Creativa y técnica
<b>Descripción</b>	<p>En esta sesión, el alumnado se enfocará en la <b>conectividad de la interfaz de la app creada en App Inventor</b>, con el objetivo de mejorar la usabilidad, la conexión y la identificación visual de los comandos. Se propondrá modificar la búsqueda por bluetooth según su función (por ejemplo, verde para buscar, rojo para detener), incorporar iconos representativos mediante la biblioteca de imágenes o recursos externos, y renombrar los botones con términos más claros o creativos (como “conectar”, “desconectar” o “buscar”). Esta fase permite trabajar el diseño digital desde una perspectiva funcional y expresiva, fomentando la accesibilidad. Además, se anima a los equipos a justificar sus elecciones gráficas y a pensar en cómo facilitar el uso de la app por parte de otros compañeros.</p> 



"Programa financiado por el Ministerio de Educación, Formación Profesional y Deportes"





# Proyecto didáctico



```
como setDisconnected
ejecutar
  poner lblStatus . Texto como " Desconectado "
  poner btnBuscar . Visible como verdadero
  poner btnDesconectar . Visible como falso
  poner btnDetener . Visible como falso
  poner btnConectar . Visible como falso

cuando Screen1 .PermisoConcedido
  nombrePermiso
  ejecutar
    si
      tomar nombrePermiso = " BLUETOOTH_CONNECT "
      entonces
        llamar Screen1 .SolicitarPermiso nombrePermiso " BLUETOOTH_SCAN "

inicializar global uartCharUuid como " 6e40002-b5a3-f393-e0a9-e50e24dcca9e "
inicializar global uartServiceUuid como " 6e40001-b5a3-f393-e0a9-e50e24dcca9e "

cuando BluetoothLE1 .Connected
  ejecutar
    poner lblStatus . Texto como unir " Conectado "
    unir lvBLE . Selección
    poner lvBLE . Visible como falso
    poner btnDesconectar . Visible como verdadero
    poner btnBuscar . Visible como falso
    poner btnConectar . Visible como falso
    poner btnDetener . Visible como falso
    llamar BluetoothLE1 .RegisterForBytes
      serviceUuid tomar global uartServiceUuid
      characteristicUuid tomar global uartCharUuid
      signed falso

cuando btnConectar .Clic
  ejecutar
    llamar BluetoothLE1 .StopScanning
    llamar BluetoothLE1 .Conectar
    indice lvBLE . ÍndiceSelección
    poner lblStatus . Texto como " Conectando.. "

cuando btnDesconectar .Clic
  ejecutar
    llamar BluetoothLE1 .Desconectar

cuando BluetoothLE1 .DeviceFound
  ejecutar
    poner lvBLE . ElementosDesdeCadena como BluetoothLE1 . DeviceList

cuando btnBuscar .Clic
  ejecutar
    llamar BluetoothLE1 .StartScanning
    poner lblStatus . Texto como " Buscando "
    poner lvBLE . Visible como verdadero
    poner btnDetener . Visible como verdadero
```



"Programa financiado por el Ministerio de Educación,  
Formación Profesional y Deportes"





```
cuando btnDetener .Clic
ejecutar
  llamar BluetoothLE1 .StopScanning
  poner lblStatus .Texto como "Detener Búsqueda"
  poner btnConectar .Visible como verdadero
```

```
cuando BluetoothLE1 .Disconnected
ejecutar
  Llamar setDisconnected
```

```
como SendText cmd
ejecutar
  llamar Microbit_Uart1 .WriteRXCharacteristic
    UART_TX
  unir
  tomar cmd
  "#"
```

A continuación, se abordará **el diseño de movimientos combinados en la micro:bit asociados cada una de las líneas de comando que se han programado**, seleccionado secuencias que integren varias acciones encadenadas, como girar y avanzar, retroceder y detener, o realizar un giro en U. Para ello, se utilizarán bloques condicionales y temporizadores en MakeCode, que permitan modular la duración de cada acción y encadenar movimientos con precisión. Además, se podrá plantear el uso de distintos tipos de interacción en App Inventor, como el evento “presionar” y “soltar” (TouchDown y TouchUp), además del clásico “click”. Esto permitirá controlar el robot de forma más dinámica, por ejemplo, haciendo que avance mientras se mantiene pulsado el botón, y se detenga al soltarlo. Esta parte de la sesión refuerza el pensamiento algorítmico, la lógica secuencial y la capacidad de ajustar el comportamiento del robot según el contexto de uso.



# Proyecto didáctico



```
bluetooth al recibir datos #
fijar message a uart bluetooth leer hasta #
si message = "A" entonces
  Parar motor ambos
  Motor ambos sentido avanzar velocidad velocidad
si no, si message = "B" entonces
  Parar motor ambos
  Motor ambos sentido retroceder velocidad velocidad
si no, si message = "C" entonces
  Parar motor ambos
  Motor izquierdo sentido avanzar velocidad velocidad
  Motor derecho sentido retroceder velocidad velocidad
si no, si message = "D" entonces
  Parar motor ambos
  Motor izquierdo sentido retroceder velocidad velocidad
  Motor derecho sentido avanzar velocidad velocidad
```



"Programa financiado por el Ministerio de Educación,  
Formación Profesional y Deportes"





# Proyecto didáctico



```

si no, si <message> = <"E"> entonces
  Parar motor ambos
si no, si <message> = <"F"> entonces
  cambiar velocidad por 5
si no, si <message> = <"G"> entonces
  cambiar velocidad por -5
si no, si <message> = <"H"> entonces
  Motor izquierdo sentido avanzar velocidad velocidad + 5
  Motor derecho sentido avanzar velocidad velocidad
si no, si <message> = <"I"> entonces
  Motor izquierdo sentido avanzar velocidad velocidad
  Motor derecho sentido avanzar velocidad velocidad + 5

```

```

al conectar bluetooth
  fijar velocidad a 20
  bluetooth servicio uart
  mostrar ícono
al desconectar bluetooth
  mostrar ícono

```



Programa financiado por el Ministerio de Educación, Formación Profesional y Deportes



GOBIERNO DE ESPAÑA

MINISTERIO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y DEPORTES





# Proyecto didáctico



	<p>Finalmente, se dedicará tiempo a la <b>documentación del proceso en una ficha visual</b>, que recoja los avances realizados, las decisiones tomadas y las dificultades superadas. Cada equipo completará una plantilla que incluya capturas de pantalla de la app personalizada, esquemas de los bloques de programación utilizados, anotaciones sobre los movimientos diseñados y reflexiones personales sobre el trabajo en equipo y el aprendizaje técnico. Esta documentación no solo facilita la evaluación y la autoevaluación, sino que también permite compartir buenas prácticas entre grupos y visibilizar el proceso creativo y técnico. Se fomentará el uso de lenguaje claro, visualizaciones sencillas y propuestas de mejora para futuras versiones del proyecto.</p>
<b>Recursos</b>	<ul style="list-style-type: none"><li>- App Inventor.</li><li>- Maqueen y placa Micro:bit</li><li>- <a href="#">Ficha completa conectividad bluetooth.</a></li><li>- <a href="#">Ficha completa movimientos combinados en MakeCode.</a></li><li>- MakeCode</li><li>- Fichas de registro.</li></ul>

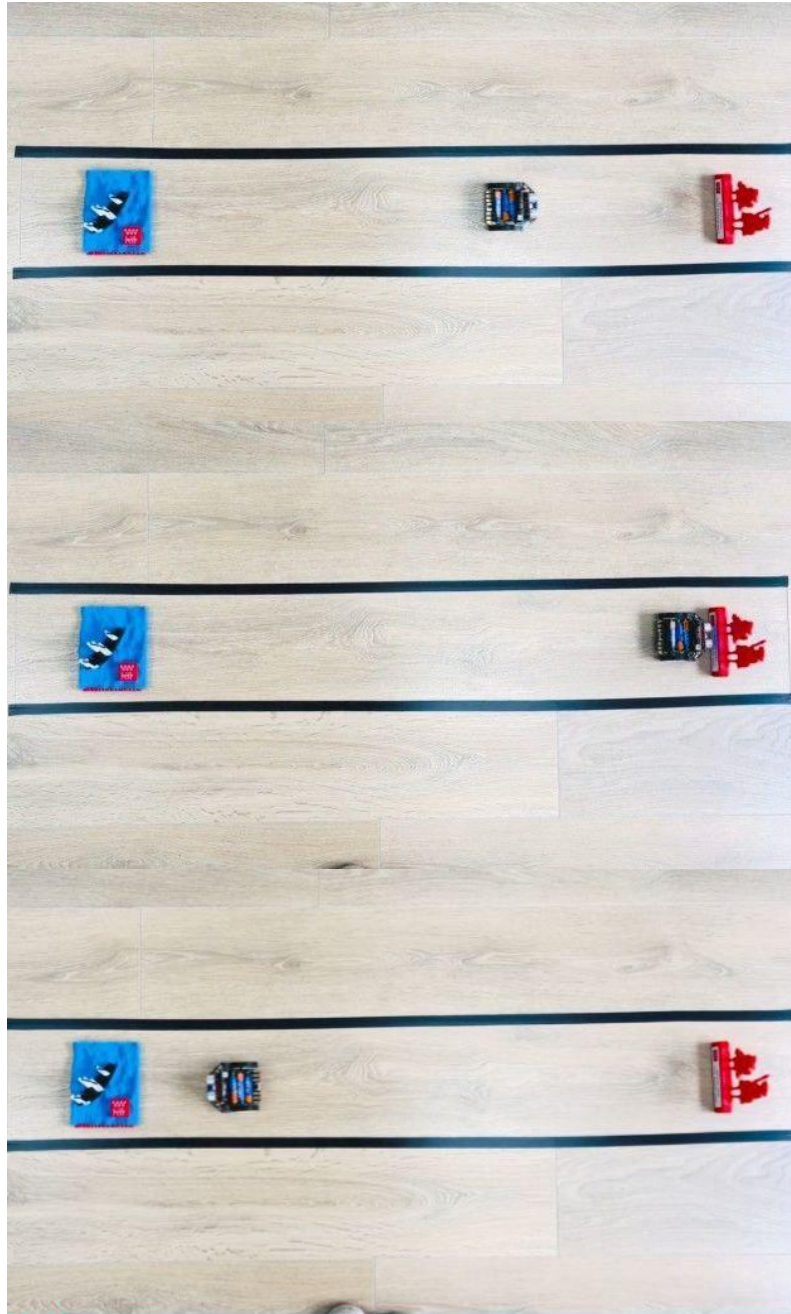
<b>Fase</b>	<b>5</b>
<b>Temporalización</b>	55 minutos
<b>Tipo de Actividad</b>	Práctica colaborativa
<b>Descripción</b>	<p>En esta quinta sesión, el alumnado pondrá a prueba sus aplicaciones y robots en una <b>simulación de recorridos o retos prácticos</b>, como laberintos, circuitos o trayectos con obstáculos. Se propondrá diseñar un espacio delimitado en el aula —con cinta adhesiva, cartulina o materiales reutilizados— que represente un recorrido que el robot Maqueen debe completar utilizando los comandos programados desde la app. Cada equipo elegirá o adaptará un reto según el nivel técnico alcanzado, pudiendo incluir giros, paradas estratégicas o trayectos en zigzag. Esta actividad permite aplicar los conocimientos adquiridos en un contexto real, reforzando la motivación y el sentido del</p>



# Proyecto didáctico

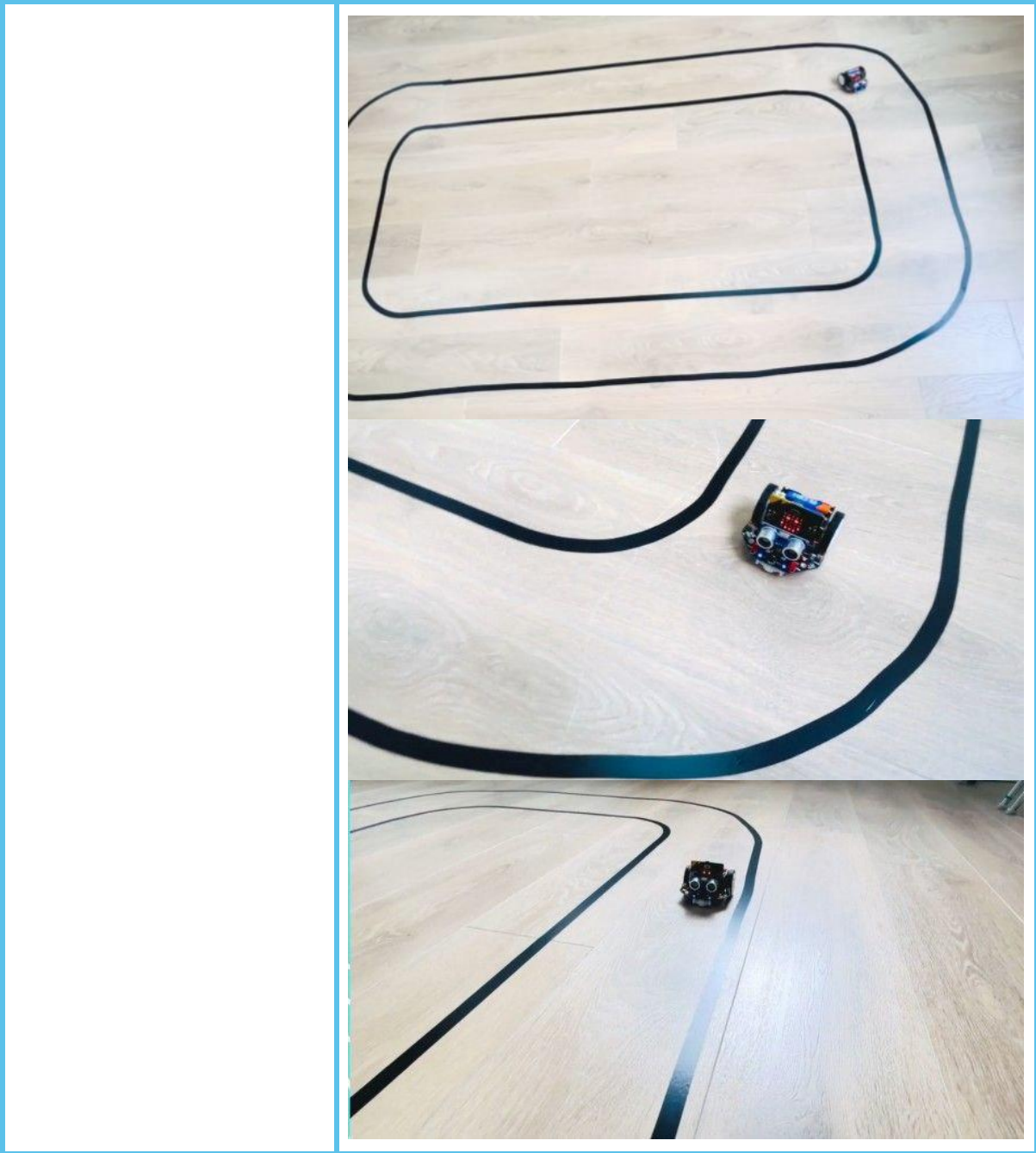


proyecto.



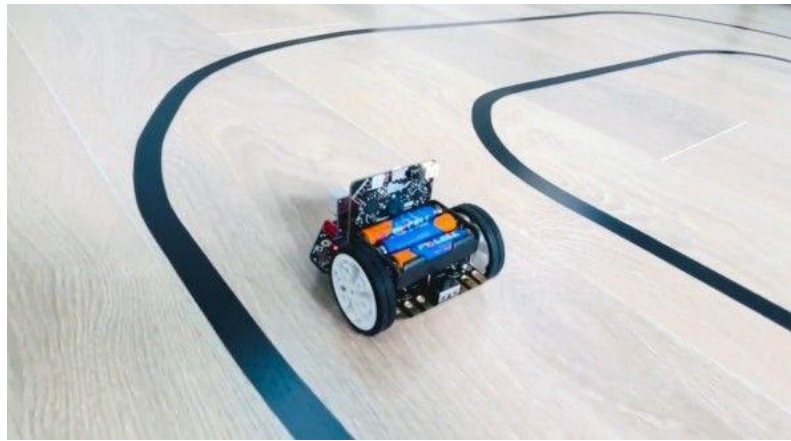


# Proyecto didáctico



"Programa financiado por el Ministerio de Educación,  
Formación Profesional y Deportes"





Durante las pruebas, se trabajará el **ajuste de velocidad, tiempos y respuesta del robot**, analizando cómo reacciona ante los comandos enviados desde la app. El alumnado observará si los movimientos son precisos, si los giros se completan correctamente y si la velocidad es adecuada para el tipo de recorrido. Se animará a modificar parámetros como la duración de los motores, la intensidad de giro o el tiempo de pausa entre acciones, utilizando bloques de temporización y control en MakeCode. Esta fase fomenta el pensamiento crítico, la mejora iterativa y la capacidad de ajustar soluciones técnicas en función de la experiencia directa.



Además, se promoverá la **observación entre equipos y el feedback cruzado**, organizando momentos en los que cada grupo pueda ver el trabajo de los demás, tomar nota de ideas interesantes y ofrecer sugerencias constructivas. Se facilitarán rúbricas de observación o fichas de evaluación entre iguales, que incluyan aspectos como funcionalidad, creatividad, precisión y comunicación. Esta dinámica refuerza la colaboración, el respeto por el trabajo ajeno y el aprendizaje entre pares, además de generar un ambiente de intercambio positivo y enriquecedor.



# Proyecto didáctico



## Recursos

- Circuito de prueba.
- Conómetros.

<b>Fase</b>	<b>6</b>
<b>Temporalización</b>	55 minutos
<b>Tipo de Actividad</b>	Expositiva y reflexiva
<b>Descripción</b>	<p>En esta última sesión, cada grupo realizará la <b>presentación final de su proyecto</b>, compartiendo con el resto de la clase la app desarrollada, los movimientos programados y las dificultades que han logrado superar. Se les animará a mostrar la interfaz personalizada, explicar cómo funciona la comunicación Bluetooth con la micro:bit y demostrar en directo los movimientos del robot Maqueen. Esta exposición permite visibilizar el trabajo técnico y creativo realizado, reforzar la expresión oral y valorar el esfuerzo colectivo. Además, se fomentará que cada equipo explique brevemente cómo resolvió los retos encontrados durante el proceso, desde errores de conexión hasta ajustes de velocidad o lógica de programación.</p> <p>Tras la presentación, se llevará a cabo una <b>autoevaluación y coevaluación</b>, utilizando rúbricas visuales y accesibles que permitan valorar aspectos como la funcionalidad de la app, la precisión de los movimientos, la colaboración en el equipo y la creatividad en el diseño. Cada alumno reflexionará sobre su propia participación, identificando fortalezas y áreas de mejora, mientras que los equipos podrán intercambiar valoraciones constructivas entre sí. Esta dinámica promueve la responsabilidad individual, el respeto por el trabajo ajeno y el desarrollo de habilidades de evaluación crítica, además de facilitar la recogida de evidencias para el docente.</p>



# Proyecto didáctico



Dimensión	Ítem evaluado	Excelente	Satisfactorio	Mejorable	Insuficiente
<b>Competencia digital y técnica</b>	Conexión y configuración Bluetooth	Conecta y configura de forma autónoma y eficaz	Conecta con ayuda puntual y comprende el proceso	Requiere ayuda constante y muestra dudas	No logra conectar ni comprender el proceso
	Programación en App Inventor	Programa con lógica clara, interfaz funcional y diseño personalizado	Programa con funcionalidad básica y diseño comprensible	Programa con errores o interfaz poco clara	No logra programar ni ejecutar acciones básicas
	Control del robot Maqueen	El robot responde con precisión a los comandos y realiza movimientos combinados	El robot realiza movimientos básicos correctamente	El robot responde de forma irregular o limitada	El robot no responde o no se ha probado
<b>Creatividad y diseño</b>	Personalización de la app	Interfaz atractiva, coherente y adaptada al reto	Interfaz funcional con algunos elementos personalizados	Interfaz poco cuidada o sin personalización	Interfaz incompleta o sin diseño
	Diseño de movimientos	Movimientos variados, creativos y adaptados al circuito	Movimientos básicos y funcionales	Movimientos repetitivos o poco eficaces	Movimientos no definidos o sin ejecución
<b>Trabajo en equipo y comunicación</b>	Colaboración en el grupo	Participa activamente, respeta roles y apoya al equipo	Participa y respeta turnos, aunque con poca iniciativa	Participa de forma irregular o con conflictos	No colabora o interfiere negativamente



Dimensión	Ítem evaluado	● Excelente	● Satisfactorio	● Mejorable	● Insuficiente
	Documentación del proceso	Registra avances, dificultades y soluciones con claridad	Registra los pasos principales del trabajo	Registros incompletos o poco claros	No hay registro del proceso
<b>Reflexión y metacognición</b>	Autoevaluación y mejora	Reconoce logros y propone mejoras con argumentos	Reconoce logros y dificultades básicas	Reflexión superficial o sin propuestas	No realiza reflexión ni reconoce el proceso
	Presentación final	Expone con claridad, seguridad y lenguaje técnico adecuado	Expone con apoyo visual y lenguaje comprensible	Expone con dificultad o sin estructura clara	No participa en la exposición o no presenta resultados

Finalmente, se dedicará un espacio a la **reflexión sobre el aprendizaje, la colaboración y el uso de la tecnología**, tanto a nivel individual como grupal. Se propondrán preguntas abiertas como: “¿Qué he aprendido sobre programación y robótica?”, “¿Qué ha sido lo más difícil y cómo lo superamos?”, “¿Cómo ha influido el trabajo en equipo en el resultado final?”, “¿Qué me gustaría mejorar o repetir en futuros proyectos?”. Esta reflexión puede recogerse en una ficha escrita, en formato mural colaborativo o mediante una dinámica oral. El objetivo es cerrar el proyecto con conciencia del proceso vivido, reforzar el valor del aprendizaje práctico y generar propuestas para futuras experiencias.

**Recursos**

- [Rúbrica de evaluación.](#)
- Espacio de exposición.
- Vídeo/fotos del proceso.



## ¿Sabías qué?

◆ Estamos trabajando con App Inventor. Explicamos cómo diseñar una app que envía comandos por Bluetooth con solo pulsar un botón.



- ◆ Estamos trabajando con micro:bit. Explicamos cómo convertirla en el cerebro de un robot que obedece órdenes desde el móvil.
- ◆ Estamos trabajando con el robot Maqueen. Explicamos cómo hacer que avance, gire o se detenga según lo que tú programes.
- ◆ Estamos trabajando con eventos táctiles. Explicamos cómo diferenciar entre “click”, “presionar” y “soltar” para controlar el robot con precisión.
- ◆ Estamos trabajando con cadenas de texto. Explicamos cómo enviar letras como “A” o “R” para que el robot sepa si debe avanzar o retroceder.
- ◆ Estamos trabajando con bloques de programación. Explicamos cómo combinar giro y avance para que el robot esquive obstáculos como un profesional.
- ◆ Estamos trabajando con la velocidad. Explicamos cómo ajustar el tiempo de giro para que el robot no se salga del circuito.
- ◆ Estamos trabajando con la observación entre equipos. Explicamos cómo dar feedback técnico y creativo para mejorar el diseño de otros compañeros.
- ◆ Estamos trabajando con documentación visual. Explicamos cómo registrar el proceso con capturas, esquemas y reflexiones para compartir lo aprendido.
- ◆ Estamos trabajando con tecnología educativa. Explicamos cómo convertir una app y una placa en una experiencia de aprendizaje interdisciplinaria.



## Evaluación

Para evaluar adecuadamente este proyecto didáctico, se han establecido procedimientos, actividades de evaluación e instrumentos que reflejan fielmente los objetivos y competencias planteados. La evaluación no solo permite medir el progreso y los logros de los estudiantes, sino que también proporciona información valiosa para ajustar y mejorar el proceso de enseñanza. A continuación, se detallan estos aspectos.

**Procedimientos**

**Actividades de Evaluación**

**Instrumentos**



**Observación directa:** durante las sesiones prácticas, se observa cómo cada alumno participa en su grupo, manipula el robot y la micro:bit, colabora en el diseño de la app y responde ante errores o ajustes técnicos.

**Intercambios orales:** se valoran las explicaciones que los alumnos dan al presentar su app, justificar decisiones de diseño, compartir dificultades superadas o proponer mejoras en las asambleas finales.

**Producciones del alumnado:** se evalúan tanto el producto final (app funcional + robot programado) como los materiales generados durante el proceso: esquemas de interfaz, bloques de código, fichas de reflexión, vídeos y presentaciones.

**Coevaluación:** se promueve la evaluación entre iguales tras las presentaciones, utilizando listas de cotejo para valorar aspectos como funcionalidad, claridad, creatividad y resolución de problemas.

**Presentación final por equipos:** cada grupo expone su proceso de trabajo, muestra la app, explica la lógica de programación y comenta cómo resolvió los retos técnicos.

**Vídeo de funcionamiento:** grabación breve donde el equipo muestra cómo se controla el robot desde la app y qué movimientos realiza. Sirve como evidencia para evaluar funcionalidad y comunicación.

**Archivo digital compartido:** carpeta en el aula virtual con capturas de pantalla, esquemas de bloques, fichas de reflexión y rúbricas cumplimentadas.

**Participación diaria:** se valora la actitud, el compromiso, la colaboración y el uso responsable del material en cada sesión del proyecto.

**Actividades prácticas:**

- Pruebas de funcionamiento del robot en recorridos simulados.
- Personalización de la app con colores,

**Lista de control:** para verificar si cada grupo ha completado los elementos esenciales del proyecto: interfaz funcional, conexión Bluetooth, movimientos programados, documentación entregada.

**Rúbricas:** para evaluar el producto final, la presentación oral, la funcionalidad de la app y la calidad del trabajo en equipo. Se adaptan por niveles de logro y se comparten con el alumnado desde el inicio.

**Listas de cotejo:** para coevaluación entre equipos tras las presentaciones, centradas en aspectos observables como claridad, originalidad, precisión técnica y resolución de problemas.

**Escalas de valoración:** para valorar la participación, la autonomía, la colaboración y el uso responsable del material durante el desarrollo del proyecto.

**Portafolio del proyecto:** recopilación de evidencias del proceso (capturas, esquemas, vídeos, reflexiones)



	<p>iconos y nombres.</p> <ul style="list-style-type: none"><li>- Diseño de movimientos combinados (giro + avance).</li><li>- Programación de eventos táctiles (click, presionar, soltar).</li></ul> <p><b>Asamblea final:</b> espacio para compartir aprendizajes, dificultades y propuestas de mejora tras la presentación de los proyectos.</p>	<p>entregadas en formato digital o físico.</p>
--	---	--



## Criterios de evaluación

- 1.1. Reconocer los sistemas tecnológicos y digitales analizando sus elementos, su funcionamiento y su impacto en el entorno.
- 2.1. Formular y resolver problemas tecnológicos aplicando el método de proyectos y estrategias de pensamiento computacional.
- 2.2. Diseñar algoritmos y programas sencillos en entornos visuales de programación.
- 3.1. Planificar, diseñar y construir productos digitales que integren hardware y software.
- 3.2. Participar en proyectos tecnológicos, cooperando en la planificación, ejecución y evaluación.
- 4.1. Usar de forma responsable dispositivos, redes y servicios digitales, aplicando medidas básicas de seguridad y protección de datos.
- 5.1. Comunicar los procesos, productos y conclusiones de un proyecto tecnológico, utilizando recursos digitales y vocabulario técnico adecuado.



## Rúbrica de evaluación para el docente (documento descargable)

	<b>Excelente</b>	<b>Satisfactorio</b>	<b>Mejorable</b>	<b>Insuficiente</b>
<b>Conexión y configuración Bluetooth</b>	Conecta y configura de forma autónoma y eficaz	Conecta con ayuda puntual y comprende el proceso	Requiere ayuda constante y muestra dudas	No logra conectar ni comprender el proceso
<b>Programación en App Inventor</b>	Programa con lógica clara, interfaz funcional y diseño personalizado	Programa con funcionalidad básica y diseño comprensible	Programa con errores o interfaz poco clara	No logra programar ni ejecutar acciones básicas
<b>Control del robot Maqueen</b>	El robot responde con precisión a los comandos y realiza movimientos combinados	El robot realiza movimientos básicos correctamente	El robot responde de forma irregular o limitada	El robot no responde o no se ha probado
<b>Personalización de la app</b>	Interfaz atractiva, coherente y adaptada al reto	Interfaz funcional con algunos elementos personalizados	Interfaz poco cuidada o sin personalización	Interfaz incompleta o sin diseño
<b>Diseño de movimientos</b>	Movimientos variados, creativos y adaptados al circuito	Movimientos básicos y funcionales	Movimientos repetitivos o poco eficaces	Movimientos no definidos o sin ejecución



<b>Colaboración en el grupo</b>	Participa activamente, respeta roles y apoya al equipo	Participa y respeta turnos, aunque con poca iniciativa	Participa de forma irregular o con conflictos	No colabora o interfiere negativamente
<b>Documentación del proceso</b>	Registra avances, dificultades y soluciones con claridad	Registra los pasos principales del trabajo	Registros incompletos o poco claros	No hay registro del proceso
<b>Autoevaluación y mejora</b>	Reconoce logros y propone mejoras con argumentos	Reconoce logros y dificultades básicas	Reflexión superficial o sin propuestas	No realiza reflexión ni reconoce el proceso
<b>Presentación final</b>	Expone con claridad, seguridad y lenguaje técnico adecuado	Expone con apoyo visual y lenguaje comprensible	Expone con dificultad o sin estructura clara	No participa en la exposición o no presenta resultados



## Atención a las diferencias del alumnado

Como docente comprometido con la inclusión y el éxito de todos los estudiantes, es fundamental adaptar las tareas y actividades para atender la diversidad en el aula. Siguiendo los principios del **Diseño Universal para el Aprendizaje (DUA)**, se pueden implementar estrategias flexibles y personalizadas que respondan a las necesidades individuales de cada alumno.

A continuación, se detallan las pautas y medidas que se va a aplicar para fomentar un entorno de aprendizaje inclusivo y efectivo:

### Organización del aula y agrupación:



Los grupos se forman de manera heterogénea, combinando perfiles técnicos, creativos y organizativos.

Los alumnos con más dificultades se ubican cerca del docente o se integran en grupos con compañeros que puedan facilitarles el acceso a las tareas.

Se asignan roles rotativos adaptados: por ejemplo, un alumno con dificultades en programación puede encargarse del diseño visual de la app o de documentar el proceso.

### **Tipos de tareas y productos adaptados:**

Todos los grupos diseñan una app funcional, pero se permite que algunos trabajen con una interfaz básica (dos botones) y otros con funciones combinadas (giro + avance).

El producto final puede presentarse como:

- Vídeo explicativo del funcionamiento del robot.
- Ficha visual con capturas de pantalla y esquemas.
- Presentación oral breve con apoyo visual. Esto permite que cada alumno contribuya desde sus fortalezas (oral, visual, técnica, organizativa).

### **Rúbricas adaptadas a los saberes básicos:**

Las rúbricas incluyen criterios como:

- Comprensión de la lógica básica de los bloques.
- Capacidad para programar al menos dos movimientos funcionales.
- Participación activa en el grupo.
- Claridad en la documentación del proceso.

Se valoran la mejora progresiva, el esfuerzo y la capacidad de explicar lo realizado, aunque el resultado técnico sea sencillo.

### **Ponderación flexible de los criterios:**

Para alumnos con dificultades técnicas, se da más peso al diseño visual, la documentación o la presentación oral.



Para alumnos con dificultades comunicativas, se prioriza la funcionalidad de la app y la implicación práctica.

Se permite entregar el producto en fases o con apoyo adicional, sin penalizar el ritmo de trabajo.

## **Refuerzo de saberes básicos durante el proyecto:**

Se ofrecen recursos visuales y tutoriales para reforzar:

- Cómo conectar la micro:bit por Bluetooth.
- Cómo usar bloques básicos en App Inventor y MakeCode.
- Cómo enviar comandos simples (“A”, “R”, “S”) y programar respuestas en la micro:bit.

Se proponen tareas intermedias como:

- Simular la app sin conexión.
- Probar solo el movimiento de avance y parada.
- Personalizar la interfaz con colores e iconos sin programar eventos complejos.

## **Ajuste del nivel de exigencia:**

Todos los alumnos deben alcanzar los saberes básicos:

- Diseñar una app con al menos dos botones funcionales.
- Programar la micro:bit para responder a comandos simples.
- Participar en la documentación y presentación del proyecto.

Los alumnos que avanzan más pueden diseñar movimientos combinados, usar eventos táctiles complejos o crear recorridos con obstáculos.

Los que necesitan más apoyo se centran en tareas esenciales, con acompañamiento y materiales simplificados.