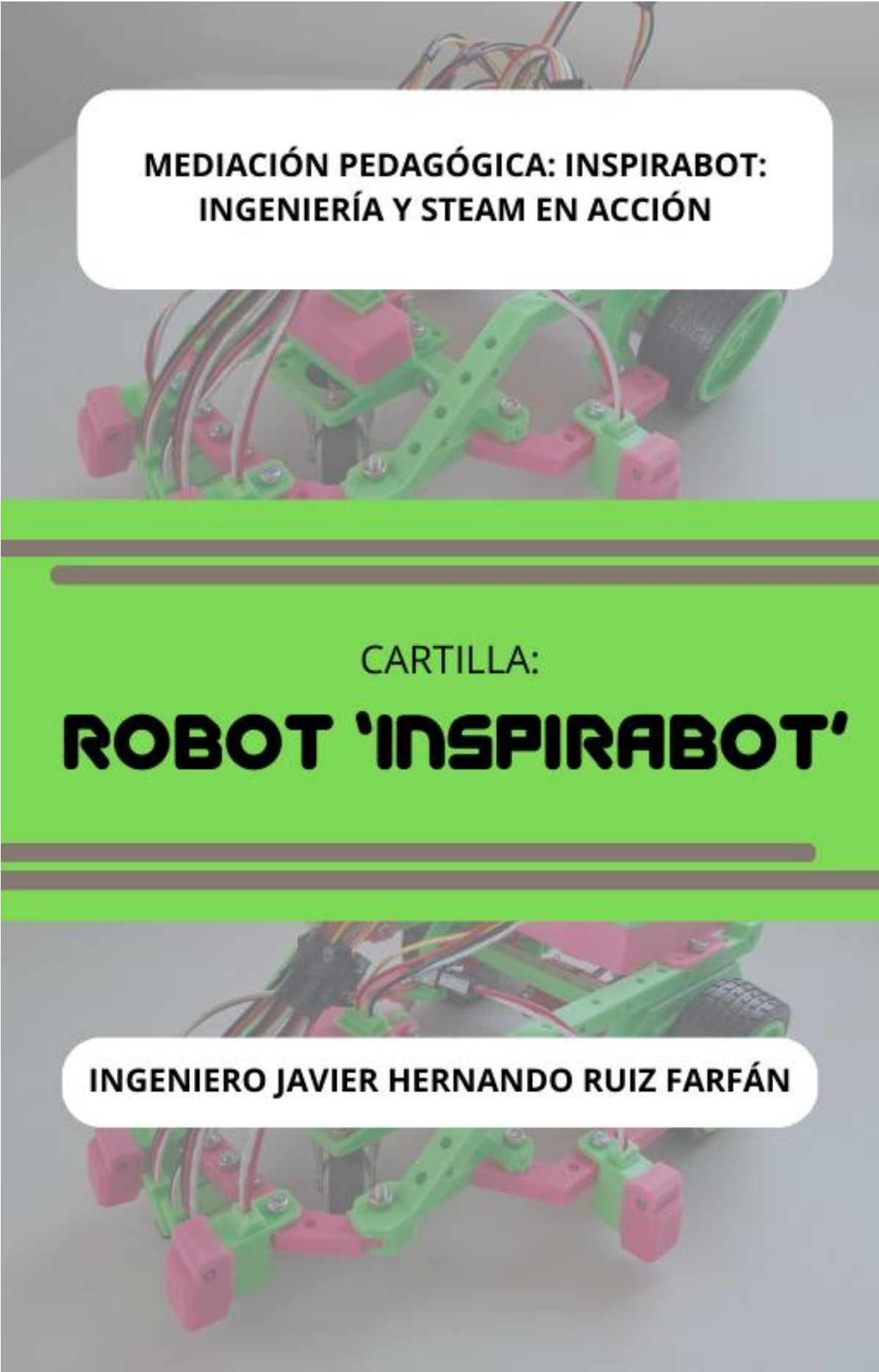




**MEDIACIÓN PEDAGÓGICA: INSPIRABOT:
INGENIERÍA Y STEAM EN ACCIÓN**

CARTILLA:

ROBOT 'INSPIRABOT'

INGENIERO JAVIER HERNANDO RUIZ FARFÁN

1. Introducción a la Robótica con mBlock y el Robot “InspiraBot”

- 1.1. ¿Qué es la robótica y por qué es importante?
- 1.2. ¿Por qué usar mBlock para programar un robot?
- 1.3. Relación con STEAM y su aplicación en el mundo real

2. Conceptos Fundamentales de Robótica

- 2.1. Sensores en el robot: Tipos y aplicaciones
- 2.2. Actuadores en el robot: Tipos y aplicaciones
- 2.3. Controladores y electrónica: Cómo funciona el cerebro del robot
- 2.4. Energía y alimentación: Consideraciones para su correcto funcionamiento

3. Conociendo el Robot “InspiraBot”

- 3.1. Descripción del robot construido
- 3.2. Componentes principales y su función
- 3.3. Cómo conectar el robot al PC y configurarlo en mBlock
- 3.4. Pruebas iniciales de conexión y comunicación

4. Fundamentos de Programación con mBlock

- 4.1. Introducción a la programación basada en bloques
- 4.2. Pensamiento lógico y estructuración del código
- 4.3. Variables, condicionales y bucles en mBlock
- 4.4. Diagramas de flujo y planificación del código

1. Introducción a la Robótica con mBlock y el Robot “InspiraBot”

1.1. ¿Qué es la robótica y por qué es importante?

La robótica es una disciplina que combina ingeniería, programación y diseño para crear máquinas automatizadas que pueden interactuar con el mundo físico. Estos sistemas pueden realizar tareas de manera autónoma o ser controlados por humanos mediante diferentes tipos de programación.

En la actualidad, la robótica tiene un impacto en diversas áreas como la industria, la medicina, la exploración espacial y la educación. En el ámbito educativo, permite a los estudiantes desarrollar habilidades STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas) a través de la experimentación y el aprendizaje basado en la práctica.

Importancia de la robótica en la educación

- Reducción de tiempos y esfuerzos humanos
 - Automatiza tareas repetitivas y físicamente exigentes.
 - Aumenta la eficiencia en sectores como manufactura y logística.
- Aumento de la precisión y la calidad

- Realiza trabajos con exactitud milimétrica, reduciendo errores humanos.
- Mejora la producción en industrias como la cirugía y la ingeniería aeroespacial.
- Automatización de tareas peligrosas
 - Opera en entornos hostiles como minería, exploración submarina y espacial.
 - Reduce la exposición humana a riesgos en la desactivación de explosivos y manejo de materiales tóxicos.
- Impulso a la innovación y el desarrollo tecnológico
 - Acelera avances en inteligencia artificial, sensores inteligentes y automatización.
 - Facilita el desarrollo de vehículos autónomos, drones y sistemas de asistencia.
- Mejora en la calidad de vida y accesibilidad
 - Robots asistenciales y prótesis mejoran la movilidad de personas con discapacidades.
 - Simplifica tareas del hogar con dispositivos como aspiradoras autónomas y asistentes inteligentes.
- Optimización en la educación y la investigación
 - Facilita el aprendizaje en ingeniería, programación y ciencias.
 - Permite simulaciones y experimentos con menor costo y sin riesgos.

1.2. ¿Por qué usar mBlock para programar un robot?

mBlock es una plataforma de programación diseñada para hacer que la robótica y la inteligencia artificial sean accesibles para todos, especialmente en la educación. Su interfaz intuitiva y su compatibilidad con diversos dispositivos la convierten en una excelente herramienta para aprender a programar un robot físico como InspiraBot de manera visual y práctica.

Ventajas de usar mBlock en la programación de robots físicos

1. Facilidad de uso con programación basada en bloques

- mBlock utiliza un entorno visual donde los comandos se organizan como bloques de colores, lo que facilita la comprensión de la lógica de programación sin necesidad de escribir código.
- Es ideal para principiantes que quieren aprender a programar robots sin conocimientos previos en lenguajes de programación.

2. Compatibilidad con hardware y sensores

- mBlock permite programar y controlar una amplia variedad de sensores y actuadores en robots físicos, como sensores ultrasónicos, motores, servos, LEDs y más.
- Esto posibilita la creación de robots con múltiples funcionalidades, desde detección de obstáculos hasta seguimiento de líneas.

3. Entorno de simulación y pruebas en tiempo real

- Se pueden simular algunos programas antes de ejecutarlos en el robot físico, lo que permite detectar errores y corregirlos sin necesidad de hardware adicional.
- La conexión entre el robot físico y mBlock permite enviar y recibir datos en tiempo real, facilitando la depuración y mejora del código.

4. Aplicaciones en proyectos STEAM

- mBlock se alinea perfectamente con el enfoque STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas), permitiendo que los estudiantes apliquen conceptos de diferentes disciplinas en proyectos creativos y funcionales.

5. Interfaz atractiva y adaptable a distintos niveles

- La interfaz de mBlock es amigable y personalizable, permitiendo que tanto principiantes como estudiantes avanzados trabajen en diferentes niveles de complejidad.

¿Cómo se usa mBlock con InspiraBot?

Para programar el robot físico *InspiraBot* con mBlock, se deben seguir estos pasos:

1. Conectar el robot al PC mediante un cable USB o conexión inalámbrica compatible.
2. Configurar el dispositivo en mBlock, seleccionando el modelo de placa y el puerto de conexión.
3. Crear el código mediante bloques en el entorno de mBlock.
4. Subir el programa al robot para que ejecute las instrucciones de manera autónoma.
5. Probar y ajustar el código para mejorar su desempeño en diferentes retos y aplicaciones.

Con mBlock, la programación de InspiraBot se convierte en una experiencia accesible, intuitiva y motivadora, ayudando a los estudiantes a desarrollar habilidades clave en lógica, pensamiento computacional y resolución de problemas de manera práctica.

1.2. Relación con STEAM y su Aplicación en el Mundo Real

La robótica educativa con mBlock y el robot *InspiraBot* no solo enseña programación, sino que también integra múltiples disciplinas del enfoque STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas). Este enfoque permite el desarrollo de habilidades clave para resolver problemas del mundo real mediante el pensamiento crítico y la creatividad.

Ciencia (Science)

- Permite explorar principios físicos, como el movimiento, la energía y la detección de objetos mediante sensores.

- Se aplican conceptos como la medición de distancias con ultrasonidos y la detección de líneas con sensores ópticos.
- Relacionado con la automatización en industrias, donde los robots realizan tareas basadas en la recolección y análisis de datos.

Tecnología (Technology)

- Uso de software de programación por bloques (mBlock) para controlar el comportamiento del robot.
- Comprensión de cómo funciona la inteligencia artificial y la automatización en procesos tecnológicos.
- Relación con dispositivos IoT (Internet de las cosas) y la robótica en el hogar, como asistentes virtuales y robots aspiradores.

Ingeniería (Engineering)

- Diseño y construcción de robots que interactúan con el entorno.
- Integración de componentes electrónicos (motores, sensores, LEDs) para crear soluciones innovadoras.
- Aplicaciones en la industria manufacturera, donde los robots optimizan la producción y la precisión en tareas repetitivas.

Arte (Art)

- Desarrollo de la creatividad en el diseño y la personalización del robot.
- Exploración de la programación para generar movimientos artísticos y animaciones con luces y sonidos.
- Relación con el arte digital y la robótica en el entretenimiento, como los robots en espectáculos y parques temáticos.

Matemáticas (Mathematics)

- Aplicación de cálculos y mediciones para programar trayectorias y movimientos precisos.
- Uso de variables y operadores matemáticos en los programas de mBlock.
- Relación con la inteligencia artificial y el análisis de datos en robótica avanzada.

Aplicación en el Mundo Real

La robótica educativa con InspiraBot no solo es una herramienta de aprendizaje, sino que también prepara a los estudiantes para comprender y desarrollar soluciones tecnológicas en la sociedad moderna.

Ejemplos de aplicación en la vida real:

- **Industria Automotriz:** Robots que ensamblan vehículos con precisión.
- **Medicina:** Robots quirúrgicos que realizan operaciones con alta precisión.

- **Agricultura:** Drones y robots autónomos que optimizan el cultivo y cosecha.
- **Servicios Domésticos:** Robots aspiradores, asistentes inteligentes y automatización del hogar.
- **Exploración Espacial:** Robots como el Perseverance en Marte, que recolecta datos científicos.

Con este enfoque, los estudiantes no solo aprenden a programar un robot, sino que comprenden cómo la robótica impacta el mundo y cómo pueden ser parte de la transformación tecnológica.

2. Conceptos Fundamentales de Robótica

En esta unidad, exploraremos los principios esenciales de la robótica, incluyendo sensores, actuadores, controladores, alimentación y la estructura mecánica del robot. Todos estos elementos trabajan en conjunto para que un robot pueda percibir su entorno, tomar decisiones y ejecutar acciones de manera eficiente.

2.1. Sensores en el Robot: Tipos y Aplicaciones

Los sensores permiten que el robot perciba su entorno al captar información como distancia, luz, sonido, temperatura, presión y contacto físico. Esta información es procesada por el controlador para que el robot pueda tomar decisiones y responder de manera adecuada.

Tipos de sensores y sus aplicaciones

1. Sensores de distancia

- **Ultrasónicos:** Miden distancias mediante la emisión de ondas sonoras. Son ideales para detección de obstáculos y navegación.
- **Infrarrojos (IR):** Detectan la presencia de objetos y pueden medir distancias cortas. Se utilizan en seguidores de línea y detección de proximidad.

2. Sensores ópticos

- **Fotocélulas:** Detectan la presencia o ausencia de luz, útiles en barreras de seguridad.
- **Sensores de color:** Diferencian colores específicos en el entorno, aplicados en clasificación de objetos y seguidores de línea avanzados.
- **LIDAR:** Utiliza láseres para mapear el entorno en 3D, empleado en robótica autónoma y vehículos sin conductor.

3. Sensores de contacto y presión

- **Bumpers o interruptores mecánicos:** Detectan colisiones con objetos.
- **Sensores de presión:** Miden fuerzas aplicadas, utilizados en robots con manos prensiles.
- **Sensores táctiles capacitivos:** Detectan contacto sin necesidad de presión física, similares a las pantallas táctiles.

4. Sensores de sonido

- **Micrófonos:** Captan sonidos y permiten el reconocimiento de voz o activación por ruido.
- **Sensores de vibración:** Detectan pequeñas oscilaciones, usados en seguridad y mantenimiento predictivo.

5. **Sensores de movimiento y orientación**

- **Acelerómetros:** Miden la aceleración del robot en diferentes ejes, útiles en sistemas de estabilización.
- **Giroscopios:** Detectan cambios en la orientación del robot, utilizados en navegación y control de equilibrio.
- **Brújulas electrónicas (magnetómetros):** Determinan la dirección del campo magnético terrestre, aplicadas en navegación autónoma.

6. **Sensores ambientales**

- **Sensores de temperatura:** Miden el calor ambiental o de componentes internos del robot.
- **Sensores de humedad:** Detectan niveles de humedad en el aire, usados en robots agrícolas.
- **Sensores de gases:** Identifican sustancias químicas o contaminantes en el aire.

7. **Sensores biométricos**

- **Sensores de pulso y ritmo cardíaco:** Usados en robots médicos o dispositivos de monitoreo de salud.
- **Escáneres de huellas dactilares:** Aplicados en robots de seguridad o control de acceso.

Cada uno de estos sensores proporciona información crucial para que el robot pueda interactuar con su entorno de manera eficiente y segura.

2.2. **Actuadores: Motores, Servos y LEDs en Acción**

Los actuadores permiten que el robot realice acciones físicas, como moverse, manipular objetos o emitir señales visuales o auditivas.

Tipos de actuadores y sus funciones

1. **Motores eléctricos**

- **Motores DC (corriente directa):** Se utilizan en robots móviles para el desplazamiento.
- **Motores paso a paso:** Permiten un control preciso del movimiento, ideales para brazos robóticos.

2. **Motores con reductores (motoreductores)**

- Reducen la velocidad y aumentan la fuerza del motor.
- Se usan para movimientos controlados con mayor precisión.
- 3. **Servomotores**
 - Motores con control preciso de posición, utilizados en articulaciones robóticas.
- 4. **Actuadores neumáticos**
 - Utilizan aire comprimido para generar movimiento, empleados en brazos robóticos industriales.
- 5. **Actuadores hidráulicos**
 - Usan líquidos a presión para generar fuerza, aplicados en robots de gran tamaño o maquinaria pesada.
- 6. **Electroimanes y solenoides**
 - Permiten agarrar o soltar objetos mediante campos magnéticos.
- 7. **Pantallas y actuadores visuales**
 - **LEDs:** Indicadores de estado o retroalimentación visual.
 - **Pantallas OLED/LCD:** Muestran información o interfaces interactivas.
- 8. **Altavoces y actuadores sonoros**
 - Reproducen sonidos o respuestas de voz en robots de asistencia.

Los actuadores permiten que el robot ejecute tareas de manera programada y adaptativa.

2.3 Controladores y Electrónica: El Cerebro del Robot

El controlador procesa la información de los sensores y envía órdenes a los actuadores.

Tipos de controladores

1. **Microcontroladores** (Arduino, ESP32, PIC)
 - Ejecutan programas básicos para controlar sensores y motores.
2. **Computadoras embebidas** (Raspberry Pi, Jetson Nano)
 - Permiten ejecutar algoritmos avanzados y procesamiento de imágenes.
3. **Controladores industriales (PLC)**
 - Se usan en entornos de automatización y robótica avanzada.

El controlador es esencial para definir la lógica de funcionamiento del robot.

2.4. Energía y Alimentación: Consideraciones para un Funcionamiento Óptimo

Fuentes de energía en robótica

1. **Baterías recargables** (Li-ion, NiMH, LiFePO4)
 - Proporcionan energía portátil para el robot.
2. **Celdas de combustible y energía solar**

- Alternativas ecológicas para robots autónomos.

3. Reguladores de voltaje y gestión de energía

- Evitan daños en los componentes electrónicos.

Una correcta gestión de energía es clave para la autonomía del robot.

2.5. Componente Mecánica del Robot

Elementos clave en la mecánica del robot

1. Chasis

- Estructura donde se montan los componentes.

2. Sistema de tracción

- Puede incluir ruedas, orugas o patas mecánicas.

3. Mecanismos de transmisión

- Engranajes, poleas o correas que transmiten el movimiento.

4. Estructuras móviles

- Brazos robóticos o mecanismos de articulación.

El diseño mecánico influye en la estabilidad y funcionalidad del robot.

3. Conociendo el Robot “InspiraBot” (Robot Físico)

En esta unidad, exploraremos la estructura del robot InspiraBot, sus componentes principales, su configuración en mBlock y las primeras pruebas de conexión.

3.1 Descripción del Robot Construido

El InspiraBot es un robot móvil con ruedas diseñado con materiales reciclables como PLA y metal. Su tamaño compacto permite una fácil manipulación y experimentación en entornos educativos.

Dimensiones:

- **Largo:** 25 cm
- **Ancho:** 22 cm
- **Alto:** 12 cm

Materiales:

- Estructura hecha con **PLA y metal reciclable**.
- Componentes electrónicos reutilizables, promoviendo la sostenibilidad.

Este diseño facilita su programación y permite realizar desafíos de navegación, detección de obstáculos y seguimiento de líneas.

3.2 Componentes Principales y su Función

El *InspiraBot* está compuesto por diferentes elementos electrónicos y mecánicos que trabajan en conjunto para su correcto funcionamiento.

Componente	Función
Controlador	Es el "cerebro" del robot, encargado de recibir y procesar instrucciones desde mBlock.
2 Motores	Proporcionan el movimiento a las ruedas, permitiendo desplazarse en distintas direcciones.
3 Ruedas	Dos ruedas motrices y una rueda loca que proporciona estabilidad y maniobrabilidad.
2 Sensores Ultrasónicos (HC-SR04)	Detectan obstáculos y miden la distancia entre el robot y los objetos cercanos.
3 Sensores QTR-1A	Permiten el seguimiento de líneas detectando contrastes entre el suelo y las líneas de referencia.
4 Sensores tipo Bumper	Detectan colisiones con objetos para evitar daños al robot.
Sensor de sonido DF Robot (analógico)	Capta sonidos para activar funciones del robot basadas en estímulos acústicos.
3 Baterías 18650	Proporcionan la energía necesaria para el funcionamiento del robot.
Tornillos y otros elementos	Permiten ensamblar la estructura del robot de manera segura.

Cada uno de estos componentes permite que el *InspiraBot* ejecute tareas como seguir líneas, evitar obstáculos y responder a estímulos del entorno.

3.3 Cómo Conectar el Robot al PC y Configurarlos en mBlock

Para programar el InspiraBot con mBlock, es necesario establecer una conexión entre el robot y la computadora. Sigue estos pasos:

Conexión del Robot al PC

1. Revisa las conexiones del robot y asegúrate de que la batería esté completamente cargada.
2. Conecta el controlador al PC mediante un cable USB o un módulo Bluetooth si es inalámbrico.
3. Enciende el robot y verifica que los componentes estén bien conectados.

Configuración en mBlock

1. Abre mBlock en tu PC.
2. Dirígete a la pestaña "Dispositivos" y haz clic en "Agregar".
3. Selecciona el controlador compatible con *InspiraBot*.
4. En la sección "Extensiones", instala las librerías necesarias para manejar los motores, sensores y la comunicación del robot.
5. Verifica la conexión: Ve a "Conectar" > "Seleccionar Puerto" y elige el puerto donde está conectado el robot.
6. Carga un programa de prueba: Un simple código para encender un LED o mover los motores puede confirmar que todo funciona correctamente.

3.4 Pruebas Iniciales de Conexión y Comunicación

Antes de comenzar a programar comportamientos avanzados, es importante probar la comunicación con el robot. Aquí hay algunas pruebas básicas:

Prueba 1: Encender un LED

- En mBlock, programa un bloque que encienda un LED del robot por 2 segundos y luego lo apague.

Prueba 2: Movimiento básico

- Programa el robot para avanzar durante 3 segundos y luego detenerse.
- Si el robot no se mueve, revisa la conexión de los motores.

Prueba 3: Lectura de sensores

- Crea un programa que muestre en pantalla la lectura de los sensores ultrasónicos y bumper.
- Acerca objetos a los sensores y verifica que los valores cambian en tiempo real.

Estas pruebas aseguran que el robot está correctamente conectado y listo para ser programado en desafíos más avanzados.

4. Fundamentos de Programación con mBlock

En esta unidad, aprenderemos los conceptos básicos de programación con *mBlock*, una herramienta que permite programar robots mediante bloques gráficos, facilitando el aprendizaje de la lógica de programación sin necesidad de escribir código complejo.

4.1. Introducción a la Programación Basada en Bloques

¿Qué es la programación basada en bloques?

Es un enfoque visual de programación en el que los comandos se representan como bloques de diferentes colores y formas, los cuales pueden unirse como piezas de rompecabezas para crear secuencias de instrucciones.

Ventajas de la programación en bloques:

- No requiere conocimientos previos de sintaxis de lenguajes de programación.
- Facilita la comprensión de la lógica y el flujo del código.
- Es ideal para la enseñanza de robótica educativa y pensamiento computacional.
- Reduce errores sintácticos, permitiendo enfocarse en la solución del problema.

¿Qué es mBlock?

mBlock es una plataforma basada en **Scratch** que permite programar robots mediante bloques gráficos y también admite programación en **Python** para usuarios más avanzados. Es compatible con diversos kits de robótica y microcontroladores.

Características de mBlock:

- Basado en Scratch 3.0.
- Compatible con Arduino y otros microcontroladores.
- Incluye simulaciones y herramientas de inteligencia artificial.
- Soporta la conversión de código en bloques a Python.

4.2. Pensamiento Lógico y Estructuración del Código

La programación requiere desarrollar **pensamiento lógico**, es decir, la capacidad de analizar problemas y resolverlos mediante una secuencia ordenada de instrucciones.

Principios del pensamiento lógico en la programación:

- **Descomposición:** Dividir un problema en partes más pequeñas.
- **Reconocimiento de patrones:** Identificar estructuras repetitivas.
- **Abstracción:** Enfocarse en la información relevante y omitir detalles innecesarios.
- **Diseño de algoritmos:** Crear una secuencia de pasos claros para solucionar un problema.

Estructuración del código en mBlock:

Un programa bien estructurado facilita la lectura y el mantenimiento del código. En mBlock, podemos estructurar el código mediante:

- **Bloques de inicio:** Definen el punto de partida del programa.
- **Bloques de acción:** Ejecutan tareas específicas, como mover el robot o encender un LED.

- **Bloques de control:** Manejan la ejecución del programa mediante condiciones y repeticiones.

4.3. Variables, Condicionales y Bucles en mBlock

Variables en mBlock

Las variables son espacios en la memoria que almacenan valores como números o texto. En mBlock, se pueden crear y modificar variables mediante bloques específicos.

Ejemplo de uso de variables:

- Guardar la cantidad de veces que un sensor detecta un objeto.
- Almacenar la velocidad de un motor para modificarla dinámicamente.

Condicionales en mBlock

Las estructuras condicionales permiten que el programa tome decisiones basadas en ciertas condiciones.

Ejemplo de condicional en mBlock:

"Si el sensor de distancia detecta un obstáculo a menos de 10 cm, el robot se detiene."

En mBlock, esto se implementa con el bloque "Si ... entonces". También se pueden usar condiciones "Si ... entonces ... sino" para definir diferentes respuestas a una situación.

Bucles en mBlock

Los bucles permiten repetir una acción varias veces o hasta que se cumpla una condición.

Tipos de bucles en mBlock:

- **Repetir N veces:** Ejecuta un bloque un número específico de veces.
- **Por siempre:** Mantiene la ejecución en un ciclo infinito.
- **Repetir hasta que:** Se ejecuta hasta que una condición sea verdadera.

Ejemplo de bucle en mBlock:

"Repetir 5 veces el parpadeo de un LED."

4.4. Diagramas de Flujo y Planificación del Código

¿Qué es un diagrama de flujo?

Es una representación gráfica de un algoritmo mediante símbolos que muestran el flujo del programa.

Símbolos básicos en los diagramas de flujo:

- **Óvalo:** Inicio o fin del programa.
- **Rectángulo:** Acción o instrucción a ejecutar.
- **Rombo:** Decisión (condicional).
- **Flechas:** Indican la dirección del flujo del programa.

Ejemplo de diagrama de flujo para un robot que evita obstáculos:

1. Iniciar programa.
2. Leer sensor de distancia.
3. Si hay un obstáculo cerca, girar.
4. Si no hay obstáculo, continuar avanzando.
5. Repetir el proceso.

La planificación del código mediante diagramas de flujo ayuda a organizar las ideas antes de programar en mBlock, facilitando la resolución de problemas.