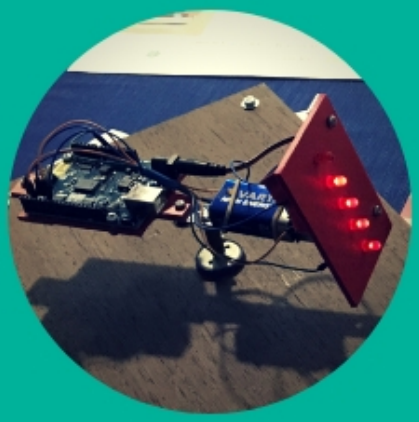
# INICIACIÓ A LA ROBÒTICA STEM

PENSAMENT COMPUTACIONAL A L'AULA

CURS ONLINE 30H MARÇ '19 - MAIG '19



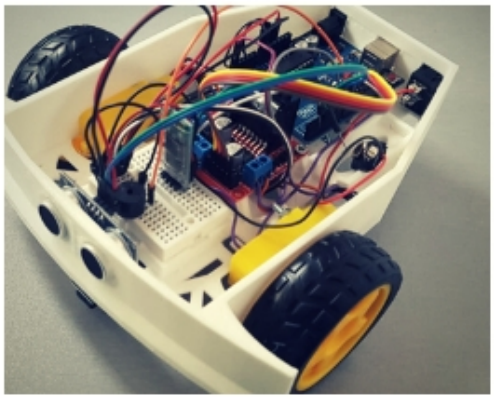
## CONTINGUTS

- Robòtica educativa i pensament computacional
- Elements d'un robot educatiu
- Revisió de robots educatius
- Plataformes per programar robots educatius
- Exemples de projectes de robòtica educativa

CURS PER A DOCENTS D'ÀMBIT CIENTÍFIC DE SECUNDÀRIA

## INFORMACIÓ DEL CURS

"**Iniciació a la robòtica STEM**" és un curs en línia de 30h coordinat pel CEFIRE CTEM per a professorat d'Educació Secundària d'assignatures d'àmbit científic.
Aquest curs suposa una primera aproximació a la robòtica educativa. Es presentaran enfocaments metodològics de com introduir la robòtica en projectes interdisciplinaris.
Revisarem diferents robots disponibles al mercat i farem especial èmfasi en els que utilitzen maquinari lliure.



## INSCRIPCIÓ AL WEB DEL CEFIRE CTEM

# TEMA 1. PARTE I: PENSAMIENTO COMPUTACIONAL

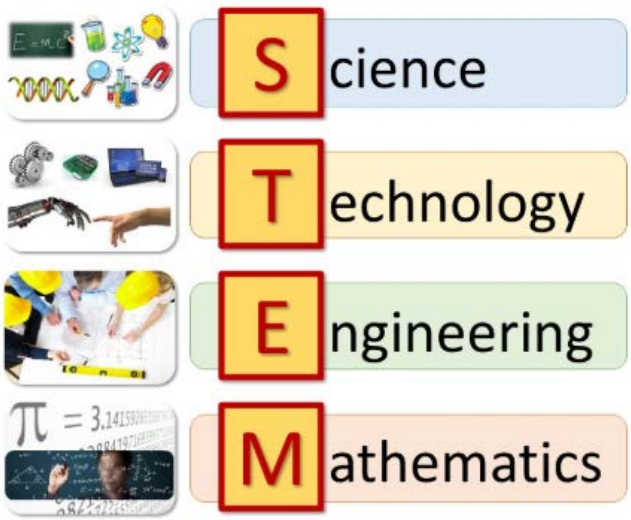
## 1. APRENDIZAJE MEDIANTE EXPERIMENTACIÓN

El aprendizaje mediante experimentación se considera una estrategia pedagógica muy importante y con un gran potencial para aumentar la motivación y participación de los estudiantes. Esto se debe a que los estudiantes suelen buscar una conexión entre el mundo (tal y como lo conocen fuera del centro educativo) y sus experiencias en el aula que los preparan para ese mundo real.

El uso de escenarios de aprendizaje que incorporan experiencias de la vida real junto con entornos tecnológicos y herramientas que ya son familiares para los estudiantes son ejemplos de enfoques que pueden aportar un aprendizaje mediante la experimentación en el aula.

## 2. ¿QUÉ ES EDUCACIÓN STEM?

Durante los últimos años cada vez ha ido cobrando mayor importancia el concepto Educación STEM que proviene de las palabras inglesas Science, Technology, Engineering y Mathematics o su equivalente en castellano CTIM, acrónimo de Ciencias, Tecnología, Ingeniería y Matemáticas. Con materias STEM los estudiantes parten de la base de la resolución de un problema a través de la creación, construcción y desarrollo de objetos.




Esta metodología es ampliamente empleada en el ámbito de la Ingeniería, y se basa en combinar recursos Matemáticos, Científicos y Tecnológicos. La Educación STEM, además de tratar las materias implicadas, tiene como objetivo llevar a cabo un proceso de enseñanza-aprendizaje de manera integrada y no como áreas de conocimiento compartimentadas. Asimismo, debe emplearse un enfoque aplicado en cuanto al desarrollo de conocimientos teóricos para su posterior aplicación práctica, orientados siempre a la resolución de problemas.

Todas las disciplinas STEM ofrecen oportunidades para desarrollar una mentalidad y un conjunto de prácticas permanente. Entre estas prácticas se desarrollan las siguientes capacidades: 1) Formular preguntas y diseñar soluciones, 2) Emplear modelos, 3) Diseñar prototipos, 4) Investigar, 5) Analizar e interpretar datos, 6) Usar el pensamiento computacional, 7) Generar un argumento a partir de la evidencia y 7) Evaluar y comunicar información.

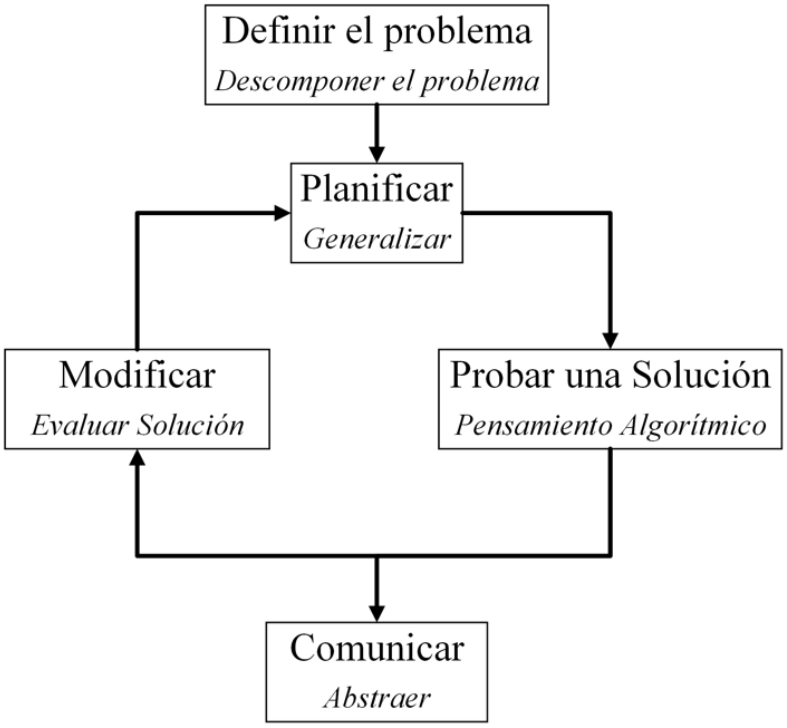
## 3. ¿Qué es Pensamiento Computacional?

Existen muchas definiciones de pensamiento computacional. Una de la más populares la propuso en 2010 la Doctora Ingeniera en Ciencias de la Computación Jeanette Wing, actualmente directora de Ciencia de la información en el Instituto Avanessians de la Universidad de Columbia, donde también es Profesora de Informática. En este sentido, Jeanette Wing define el pensamiento computacional como el "*proceso mental utilizado para formular problemas y sus soluciones de forma que las soluciones se plasman de manera adecuada para ser llevadas a cabo por cualquier agente que procese información*". Esta definición, implica que el pensamiento computacional es un proceso de pensamiento independiente de la tecnología. Mediante el pensamiento computacional somos capaces de diseñar soluciones para ser ejecutadas por un ordenador, una tableta, un humano, o cualquier combinación de ellos.

## 4. Habilidades Empleadas en el Pensamiento Computacional

Las habilidades empleadas en el pensamiento computacional pueden definirse como:

- abstracción es la habilidad para conceptualizar una idea.
- descomposición es la habilidad empleada para simplificar un problema en partes más pequeñas y sencillas de resolver.
- generalización es la habilidad para reconocer patrones, es decir, para identificar las partes de una tarea que se han trabajado previamente.
- pensamiento algorítmico es la habilidad para crear una serie ordenada de pasos con el propósito de resolver un problema.
- evaluación es la habilidad empleada para verificar si un prototipo o solución funciona correctamente y, si no ocurre de esta forma, es también la habilidad para identificar qué parte debe mejorarse.




*Esquema sobre los pasos para conseguir la resolución del problema*

# TEMA 1. PARTE II: ROBÓTICA EDUCATIVA

## 5. ¿QUÉ ES LA ROBÓTICA EDUCATIVA?

La Robótica Educativa puede entenderse como una disciplina que permite a los estudiantes concebir, diseñar y desarrollar robots mediante el empleo de kits que incluyen diferentes elementos mecánicos y electrónicos que pueden ser programados.

Sin embargo, la Robótica Educativa también puede ser tratada como una metodología didáctica en la que el robot se emplea como una herramienta para resolver una problemática concreta. En este sentido, se suelen emplear dinámicas basadas en la creatividad y la innovación, de modo que se les propone un reto a los alumnos (que puede ser tecnológico o de otro ámbito como científico, matemático, musical…) y ellos deben resolverlo mediante el montaje y programación del robot.

## 6. ¿POR QUÉ EMPLEAR ROBÓTICA EDUCATIVA EN EL AULA?

La Robótica Educativa, además de posibilitar la creación de comportamientos y poder programarlos en el propio robot, posee otras ventajas al emplearla como herramienta docente debido a que facilita la adquisición de conocimientos de modo lúdico, basándose en el trabajo colaborativo, y del desarrollo del pensamiento lógico y computacional, integrando, además, el enfoque pedagógico STEM y el uso de la programación en bloques.

La Robótica Educativa no se trata exclusivamente de que el docente enseñe robótica, sino de que utilice este recurso tecnológico en su asignatura como factor de motivación para, a partir del interés, llevar al alumno a la construcción de su propio conocimiento y al desarrollo de competencias como: la autonomía, la iniciativa, la responsabilidad, la creatividad, el trabajo en equipo, la autoestima y el interés por la investigación.

En el siguiente video titulado “canalTIC: Robótica Educativa” en el que Ana Alegría Blázquez. Técnico del Gabinete de Tele-educación de la UPM, se exponen los orígenes de la Robótica Educativa y los posibles beneficios de emplear robots en el aula:


https://youtu.be/bBjw0_swTxQ

## 7. Introducir la Robótica Educativa en Disciplinas STEAM

Un robot educativo representa una herramienta docente muy interesante en tanto que puede emplearse para trabajar aspectos de diferentes ámbitos además del tecnológico. La versatilidad que proporciona el poder incluir diferentes bloques funcionales y definir el comportamiento del robot ofrece la oportunidad de emplearlo para trabajar disciplinas como las matemáticas, ciencias, música…

Algunos robots cuentan con altavoces que pueden emitir sonidos de forma que los alumnos pueden programar diferentes notas musicales para crear una determinada sinfonía, combinando los conceptos de educación musical, tecnología y computación. Dentro de este proyecto se pueden trabajar la interconexión del microprocesador con el altavoz, el empleo de los diferentes bloques que permiten la reproducción de cada nota y los conceptos relacionados con los tonos y tiempos de reproducción de los mismos.

En el ámbito de las ciencias, es posible utilizar diferentes sensores para estudiar diferentes magnitudes como la temperatura, humedad, luminosidad… De este modo, es posible llevar a cabo proyectos como la construcción y monitorización de un invernadero o de un acuario el análisis de los diferentes parámetros.

En cuanto a la combinación de la Robótica Educativa con otros ámbitos más alejados de las disciplinas STEM un ejemplo puede ser la combinación del robot con el área de la literatura. Podría implementar un robot que fuera programado para plantear diferentes preguntas relacionadas con dicha disciplina. De esta forma se puede programar el robot para que plantee preguntas por una pantalla y que se pueda interactuar con él mediante una serie de botones para indicar cuál es la respuesta a dicha pregunta.

Mediante el siguiente enlace, es posible acceder al video titulado “Programar para aprender sin límites | Antonio García Vicente | TEDxYouth@Valladolid”. En esta charla Antonio García, un niño de ocho años, nos muestra diferentes proyectos tecnológicos en los que se combina la programación y la Ingeniería con diferentes disciplinas como la Ciencia o la Música:


https://youtu.be/9hUjhIfs-bw

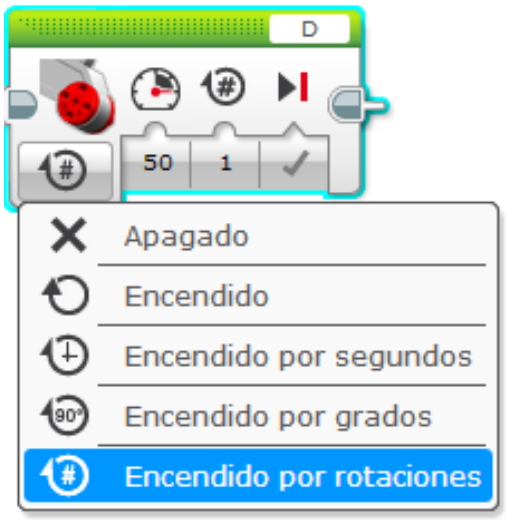
## 8. Ejemplo de Uso de la Robótica Educativa combinada con las Matemáticas

A continuación, se va a definir una actividad basada en el empleo de un robot educativo para trabajar conceptos relacionados con el ámbito de las matemáticas, concretamente se van a trabajar conceptos relacionados con el empleo del número pi y magnitudes para determinar distancias. De este modo, es posible programar el robot para que avance la longitud deseada a partir de determinar cuántos centímetros avanza el robot por cada rotación completa de las ruedas.

El enunciado de forma simplificada de la actividad propuesta sería el siguiente: "*Consigue que el robot avance en línea recta una distancia de 50 cm*". En la Figura se muestra una representación del ejercicio propuesto:


*Representación del enunciado de la actividad.*

Aunque parece una tarea sencilla, la complejidad de este enunciado radica en que los bloques para programar un movimiento del robot no permiten introducir centímetros, sino grados de giro, rotaciones completas o parciales o segundos, como muestra la siguiente imagen.

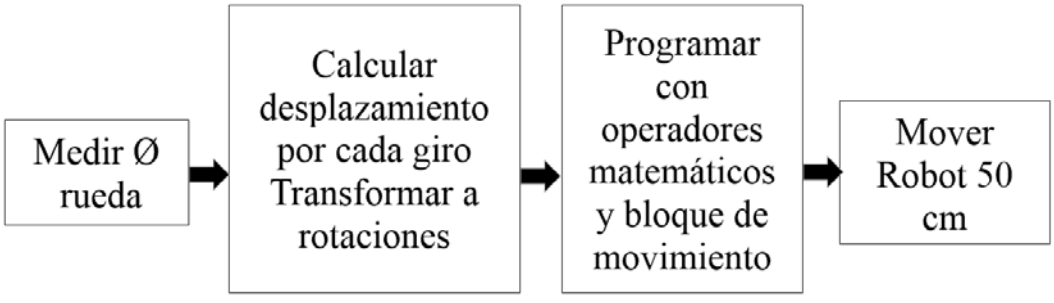



*Bloque para programar un movimiento del robot.*

Por tanto, es necesario medir el diámetro de la rueda del robot y calcular el perímetro de dicha circunferencia para conocer los centímetros que se desplaza el robot en línea recta cuando las ruedas realizan una rotación. De este modo, una vez planteado el problema, siguiendo el esquema para resolver retos expuesto en la parte de Pensamiento Computacional, en primer lugar, se descompone en partes más pequeñas:

- Medir rueda: es necesario medir la rueda para el cálculo del desplazamiento lineal del robot respecto al número de rotaciones de las ruedas.
- Operaciones matemáticas: se necesitan realizar diferentes cálculos para determinar el número de rotaciones que debe realizar el motor para conseguir la distancia de 50 cm para introducirlas en el bloque de movimiento del programa.
- Programar: Crear un código para que el robot realice el comportamiento especificado.
- Mover el robot 50 cm: Saber representar 50 cm, para verificar si el robot ha avanzado la distancia deseada.

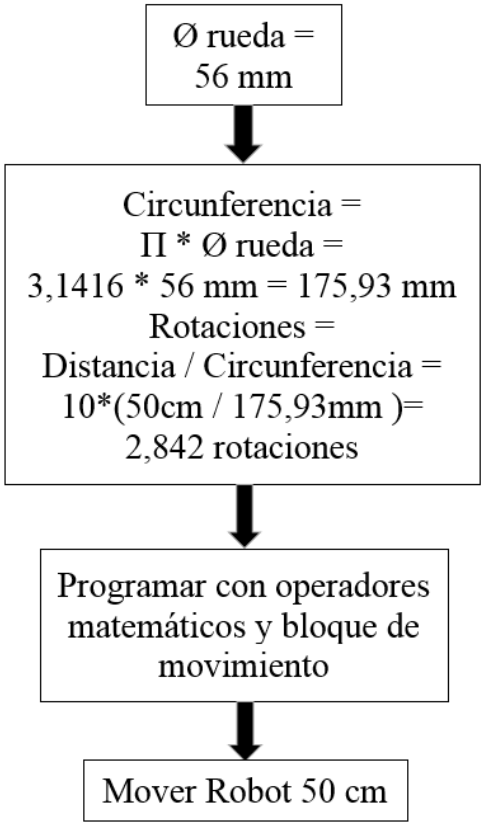
El siguiente paso se corresponde con la organización y planificación de las partes obtenidas a partir de la descomposición, de modo que se debe pensar en la secuencia de funcionamiento del robot, que puede ser representada como se muestra en la siguiente figura.




*Planificación de las tareas a realizar.*

Si el trabajo se realiza en grupos, se podría asignar la resolución de cada una de estas tareas específicas a cada uno de los miembros para fomentar el trabajo colaborativo.

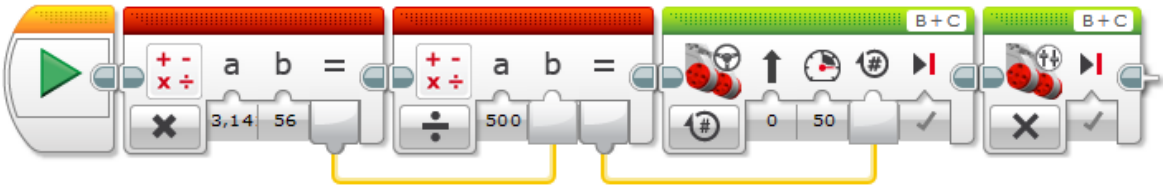
Seguidamente, el alumno debe emplear el pensamiento algorítmico y llevar a cabo las operaciones matemáticas necesarias para resolver el problema. A continuación, deberá traducir las subtareas en el código de bloques que debe programar en el robot. Por tanto, deberá de haber recibido algunas lecciones sobre los bloques que posee el entorno de programación del robot, para poder traducir las subtareas en bloques o combinación de bloques. En el siguiente diagrama se muestran los pasos y cálculos a desarrollar para conseguir resolver el problema.




*Planificación de las tareas a realizar, aplicando el pensamiento algorítmico.*

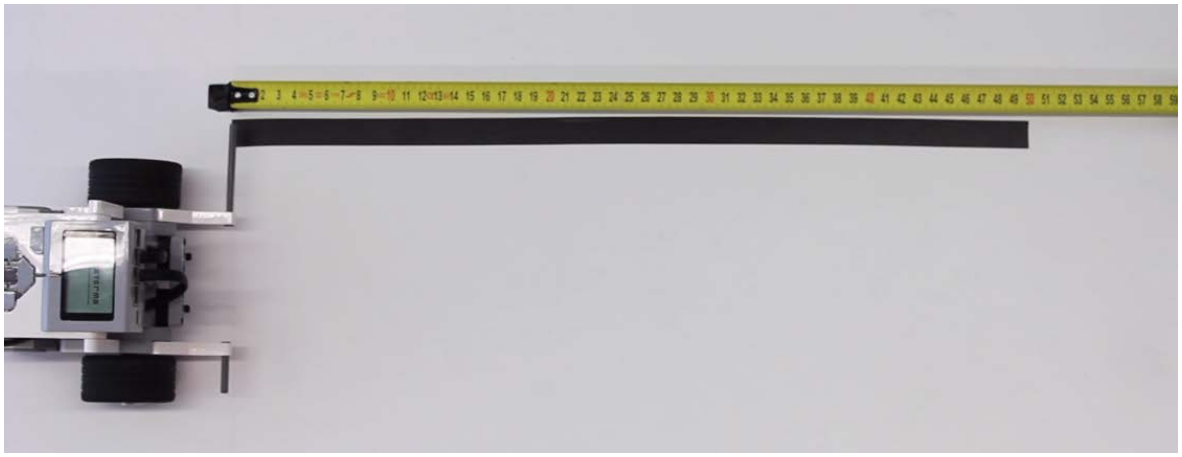
Tomando como ejemplo uno de los entornos de programación de LEGO, el programa resultante a realizar para que el robot avance 50 cm se resolvería con los bloques que se muestran en la siguiente captura. En este programa se ha incluido un primer bloque que indica el inicio del programa (bloque con el símbolo reproducir). A continuación, el segundo bloque realiza el producto para obtener la circunferencia de la rueda (3,14 x 56 cm de diámetro de la rueda del robot), que corresponde con la distancia que el robot avanza de forma lineal por cada rotación completa de la rueda. A continuación, en el tercer bloque se ha divido la distancia total que debe recorrer (500 mm) el robot entre el valor del perímetro de la circunferencia de la rueda (que ha sido calculado en el segundo bloque). El valor

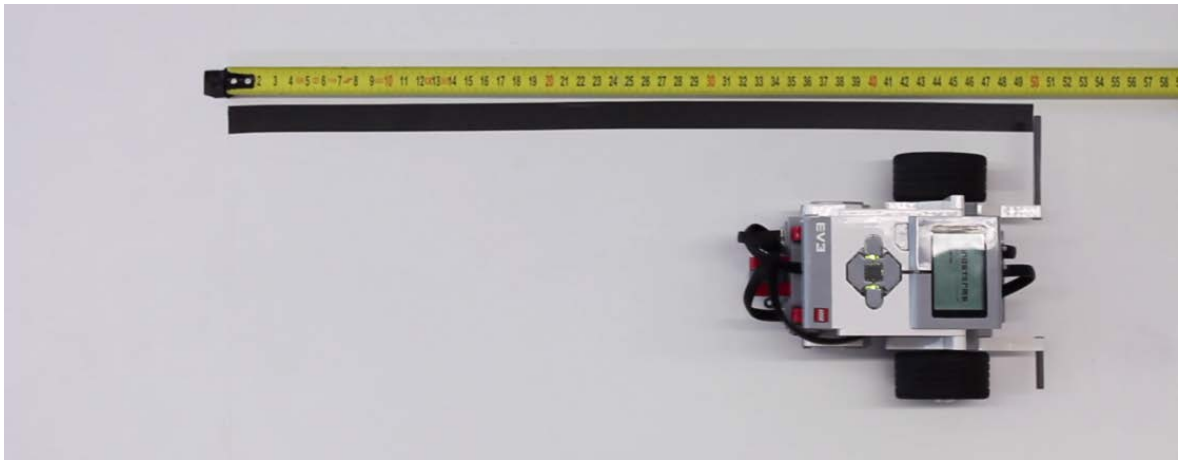
obtenido de la división realizada en el tercer bloque, se ha introducido en el bloque de movimiento del robot (cuarto bloque) para que avance hacia delante con una potencia de 50. El último bloque indica que, una vez se hayan avanzado los 50 cm (o 2,842 rotaciones calculadas con los bloques dos y tres), el robot se parará.



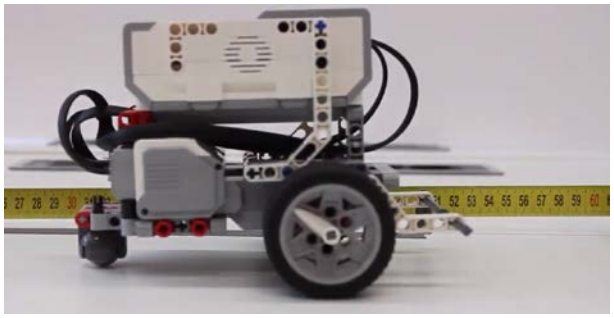

*Resolución de la actividad empleando bloques de programación*

En las siguientes imágenes se muestra la posición origen del robot y la posición final después de avanzar las rotaciones calculadas para avanzar 50 cm.





En el siguiente link se puede ver el clip de vídeo en el que se muestra el comportamiento programado:



https://youtu.be/E5qvTD9qPXY

Como puede observarse, la línea negra tiene una longitud de 50 cm, por lo que el robot, al estar correctamente programado, avanza exactamente dicha distancia.

Con esta actividad es posible probar diferentes alternativas para ver qué ocurriría si se tomaran menos decimales del número pi (la longitud avanzada no sería tan precisa) o, por ejemplo, si en lugar de introducir la distancia a recorrer en milímetros se introdujera en cm (al introducir el valor 50 en lugar de 500 el robot avanzaría únicamente 5 cm).

# TEMA 2.1. ELEMENTOS DE UN ROBOT EDUCATIVO I

## 1. DESCRIPCIÓN DE UN ROBOT.

### 1.1 ¿QUÉ ES LA ROBÓTICA?

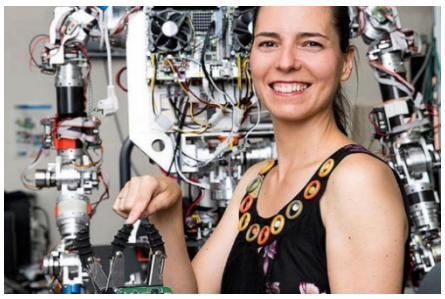
Antes de definir lo que es un Robot vamos a dar una definición básica de los que es la Robótica la cual podemos definir como:

- Es el área de la Inteligencia Artificial (IA) que se encarga de los estudios de los robots.
- La robótica no sólo incluye elementos de IA sino también de mecatrónica, computación y otras áreas de la Ingeniería.

Tambien vamos de definir Inteligencia Artificial (IA) como:

- "El estudio de cómo lograr que las computadoras realicen las tareas que, por el momento, los humanos hacen mejor." Rich, Knight, 1991.

- "La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente." Luger y Sutublfiel, 1993.



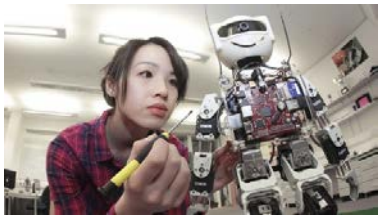
Concepción Alicia Monje Micharet

Premio Mujer y tecnología.

### 1.2 ¿QUÉ ES UN ROBOT?

La palabra robot procede de la palabra checa "robota" que literalmente significa "esclavitud", "servidumbre forzada".
Si damos una definición más técnica lo, podemos definir como:

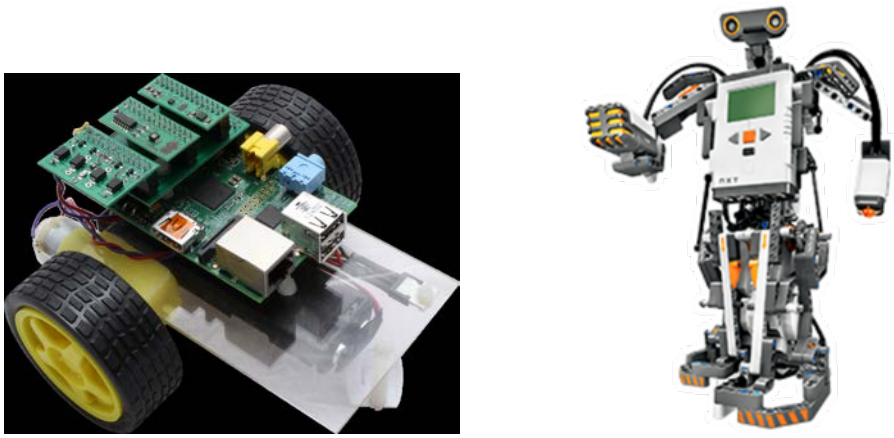
- Máquina mecánica o autómata capaz de interactuar con el entorno y tomar decisiones propias.



## 1.3 CARACTERÍSTICAS DE UN ROBOT.

Normalmente un robot se suele componer de las siguientes partes:

- Mecanismo para desplazarse.
- Mecanismo para percibir el mundo exterior.
- Mecanismo para interactuar con el entorno.
- Mecanismo de procesamiento de la información.



Distintos tipos de Robots Eductivos.

Esto no quiere decir que un robot pueda prescindir de culaquiera de estos mecanismos si no fuese necesario para la tarea para la cual fuese diseñado.

Por lo tanto podemos definir el esquema de bloques y de funcionamiento de un robot mediante el siguiente figura.

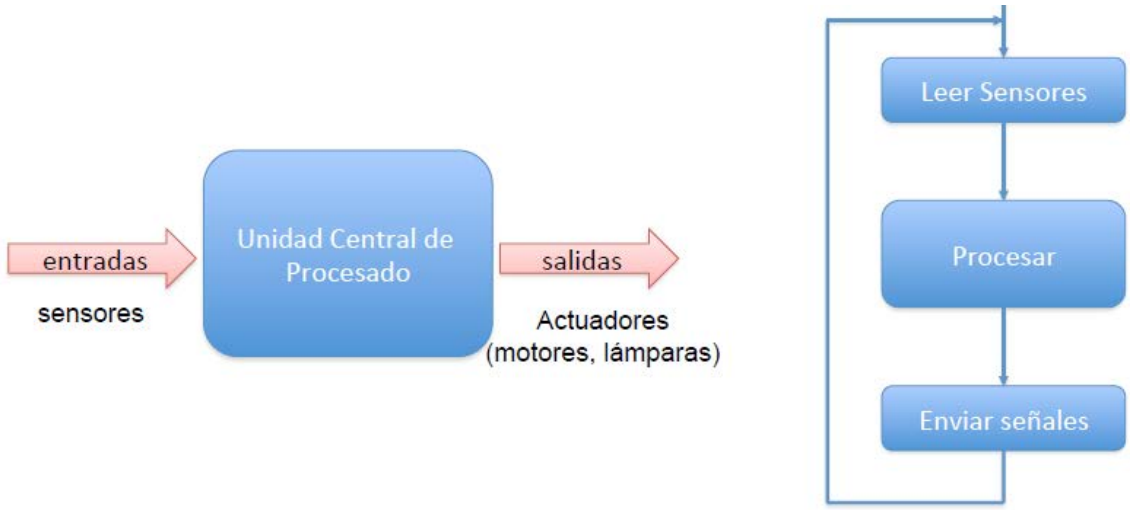



Diagrama de un Robot.

En el siguiente aparatados vamos a profundizar en los distintos bloques del Robot.

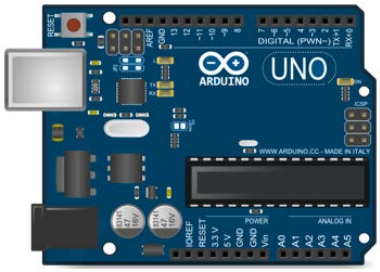
## 2. UNIDAD DE CONTROL.

Es el mecanismo en el cual se procesará toda la información de nuestras entradas (sensores) y activando las salidas pertinentes (actuadores).

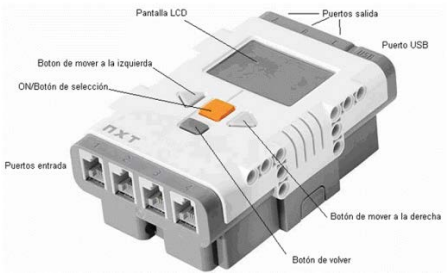
Por lo tanto, podemos decir que la unidad de control será el **cerebro** de nuestro Robot y será el sistema en cual el usuario o usuaria guardaría todas las instrucciones (código) necesarias para que nuestro robot realice la tarea deseada.

En los Robots educativos las unidades de control están basadas principalmente en microcontroladores, ya que poseen las siguientes características idóneas para estos sistemas electrónicos.

- Microcomputador en un único chip.
- Tamaño reducido.
- Bajo consumo.
- Funcionamiento por baterías.
- Integración de Entrada/Salida.
- Herramientas de desarrollo.
- Plataformas open source.




Unidad de control basada en Arduino.




Unidad de control NXT lego.

Los parámetros de interés de las unidades de control basadas en microcontroladores son los siguientes:

- Ancho de bus (número de bits 8, 16, 32,..).
- Frecuencia del reloj (10MHz .. 100MHz).
- Memoria de datos y programa.
- Número de señales digitales I/O.
- Conversión AD - DA.
- Modulación PWM.
- Temporizadores.
- Comunicaciones (UART, I2C, SPI,..).
- Interrupciones.

## 3. SENSORES (ENTRADAS).

Podemos definir los sensores como dispositivos capaces de convertir una magnitud física en una señal eléctrica.

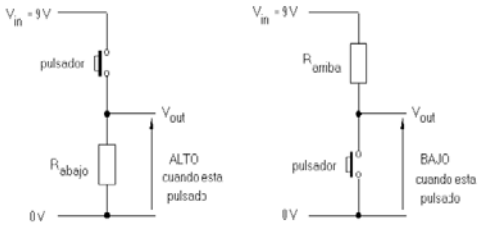
Se dispone de un gran abanico de opciones es decir de magnitudes físicas a sensar y principios de adquisición

En los siguientes aparatados vamos a repasar los sensores típicos y más utilizados en la Robótica educativa.

### 3.1. SENSOR DE CONTACTO.

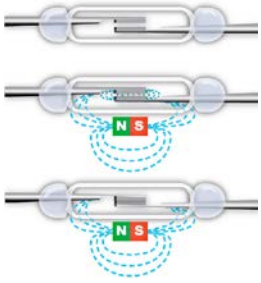
Estos sensores están basados en pulsadores y tienen un conexionado y un uso muy simple, activado o no activado además de ser muy robustos.
Si lo asemejáramos a algún sentido podríamos decir que proporcionas el **sentido del tacto** a nuestros robots.

Sensores de contacto.




Esquema eléctrico de montaje.

Tambien tenemos que decir que existen pulsadores sin contactos como en el caso de los contactos Reed que se activan mediante la aplicación de un campos magnetico.




Funcionamiento de contacto Reed.

Un ejemplo claro de la utilización de estos sensores se puede ver en los famosos robots aspiradoras, que casi todo el mundo tiene en casa, si observamos cuando el robot colisiona con una pared u otro objeto de la casa este cambia de dirección automáticamente.

Para conseguir que el robot cambie de dirección simplemente colocamos sensores de contacto alrededor del robot los cuales, al ser pulsados, por una colisión, indicaran al robot que tiene un obstáculo en la posición del sensor presionado y que por lo tanto deberá ejecutar la maniobra para evitar el obstáculo pertinente teniendo en cuenta los sensores o sensor presionado.

Robot aspirador evitando una pared.

En el siguiente enlace podemos ver la utilización de estos sensores en un robot que es capaz de detectar si colisiona con un obstáculo y variar su trayectoria en consecuencia.

https://www.youtube.com/watch?v=5_FVXeYAEso

## 3.2. SENSOR DE LUZ/COLOR.

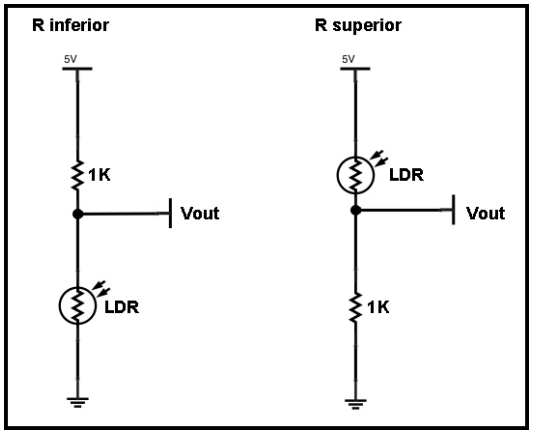
Estos sensores varían su salida dependiendo de la variación de la intensidad de la luz.

Según su fabricación podemos dividirlos en dos grandes grupos:

- Basados en LDR (Light Dependant Resistor)

  - Son simples y robustos.
  - Una gran gama con lo cual podemos sensar desde el espectro IR hasta el UV.
  - Son analógicos por lo tanto necesitaremos un conversor AD (Analógico/Digital) para que nuestro procesador pueda interpretar los valores.
  - Son de respuesta lenta por lo que tendremos que esperar un tiempo mínimo para obtener unas medidas fiables.
  - Los trataremos como resistencias variables con la intensidad de la luz.

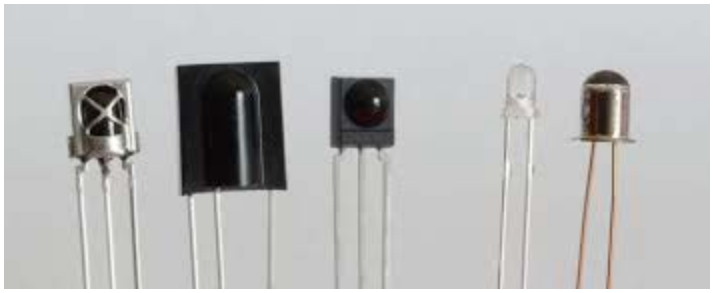
Sensor de luz basado en LDR.




Esquema de montaje tipico del sensor de luz.

- Basados en fotodiodo.

  - Sensibles a espectro más ancho.
  - Son de respuesta mucho más rápida que los basados en LDR.
  - Necesitan de un circuito de acondicionamiento, aunque pueden llevarlo ya integrado.
  - Los podemos conseguir con salida analógica o digital.



Sensores basados en fotodiodos.

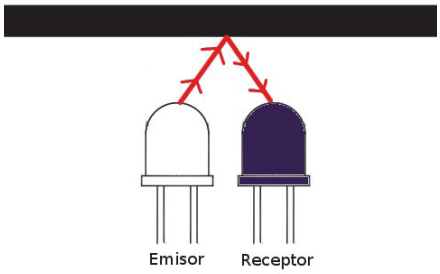
Un ejemplo típico de utilización de estos sensores en la Robótica Educativa es fabricar un robot que se active cuando se le enfoca con una luz.

En el siguiente enlace podemos ver la fabricación de un Robot seguidor de luz de manera muy sencilla.

https://www.youtube.com/watch?v=Liqf2jS_hsk

Otro ejemplo de utilización seria el control de toldo que se abriría cuando es de día y se cerraría cuando es de noche.

Tenemos un caso especial de estos sensores los cuales están formados por un emisor de luz y un receptor lo que nos permite medir tanto luz ambiente como luz reflejada.




Esquema sensor de color.

Esta configuración del sensor es muy útil ya que nos permitirá detectar colores.
Por lo tanto, por cada fuente de luz podremos diferenciar un color, si tenemos una fuente RGB (fuente de luz de varios colores) podríamos distinguir hasta 6 colores distintos.

El funcionamiento es sencillo:

Como todos sabemos el cuándo percibimos un color los que realmente está sucediendo es que el objeto absorbe el espectro de los distintos colores menos el espectro del color que percibimos, ya que este se refleja y es captado por nuestros ojos.

De manera análoga trabajamos con este sensor, ya que emitimos el espectro de un color determinado y “sensamos” el valor en el receptor si este valor es elevado sabremos que ese espectro de luz se ha reflejado, por lo tanto, la superficie u objeto será de ese color.

El ejemplo típico de utilización de estos sensores es la creación de un robot seguidor de líneas ya que gracias a estos sensores podremos posicionar nuestro robot sobre la línea.

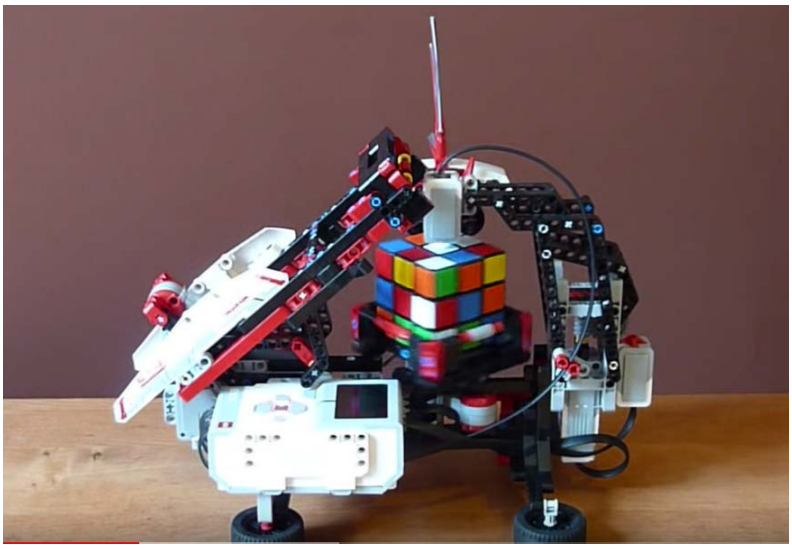
Al final lo que hacemos es capturar la luz reflejada ya que esta nos indicara si el sensor está posicionado sobre la línea o está fuera de esta para la unidad de control actué en consecuencia sobre los motores, ya que la línea es de color diferente al suelo (Notar que cuando más alejados estén los colores en la paleta cromática mejor detectaremos si nos hemos salido ya que las variaciones en el sensor serán mayores).

El siguiente video nos muestra la utilización del Robot educativo NXT como un seguidor de líneas básico.

https://youtu.be/m4bmw_iJdSo

Otro ejemplo de la utilización del sensor de color es un robot que resuelva el cubo de Rubik.

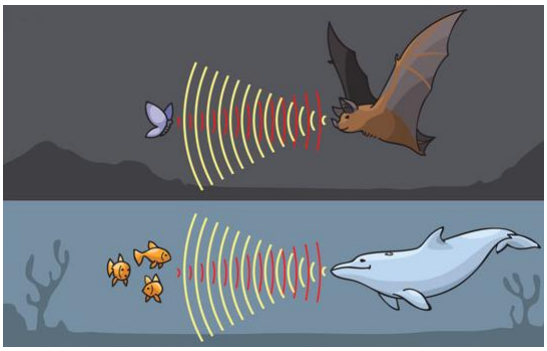
El siguiente enlace veremos cómo se resuelve le cubo de Rubik mediante el Robot educativo EV3.



https://www.youtube.com/watch?time_continue=34&v=MQvG6IgrgYQ

## 3.3. SENSOR DE DISTANCIA.

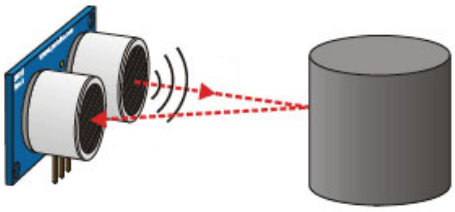
Estos sensores están basados en los ultrasonidos, su funcionamiento es parecido al de un sonar o al mecanismo que utilizan los murciélagos para "ver", por lo tanto, podríamos decir que son el **sentido de la vista** de nuestro robot.



Visón por ultrasonidos.

Este sensor nos proporcionar el tiempo que tarda en volver la señal emitida al receptor y para obtener la distancia se calcula **el tiempo de vuelo**, mediante la fórmula mostrada en la siguiente imagen.

Normalmente no tendremos que calcular el tiempo de vuelo ya que las propias aplicaciones que se utilizan para programar los diferentes robots ya se encargan de generar el código necesario para el cálculo de la distancia, por los tanto el cálculo del tiempo de vuelo suele ser transparente para el programador y facilitando el uso de estos sensores para los menos expertos.

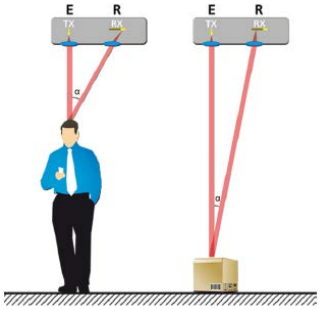


Tiempo = 2 * (Distancia / Velocidad)
Distancia = Tiempo · Velocidad / 2

Calculo de tiempo de vuelo.

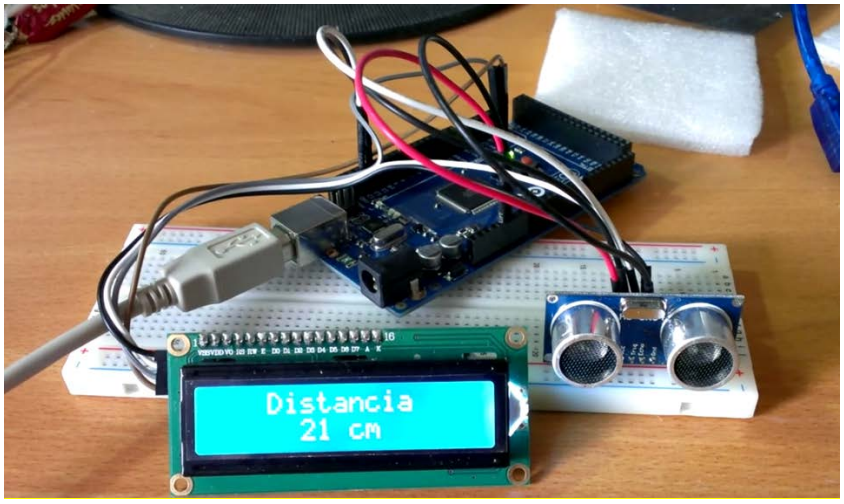
Sensor de Ultrasonidos.

Aunque es menos común tenemos sensores de distancia basados en infra-rojos los cuales trabajas de forma muy parecida a los sensores de ultrasonidos, pero estos en vez de calcular en tiempo de vuelos, calculan la distancia mediante el Angulo de reflexión.




Esquema de funcionamiento de sensor por infra-rojos.

En el siguiente enlace se podrá ver la utilización del Robot educativo mBot y un sensor de ultrasonidos mediante los cuales se puede representar por una pantalla la distancia del sensor a los obstáculos.




https://www.youtube.com/watch?v=ti8u82gz1Kk

## 3.4. SENSOR DE SONIDO.

Estos sensores son conocidos como micrófonos y mediante ellos seremos capaz de sensar la presión sonora que te tenemos en el ambiente, si lo comparamos con los sentidos de un ser humano podríamos decir que son los oídos de nuestro robot.

Un micrófono es un dispositivo que permite transformar las vibraciones del sonido en señales eléctricos.

Una vez transformadas en señales eléctricas podremos trabajar con ellas amplificándolas o modificándolas según nos interese.

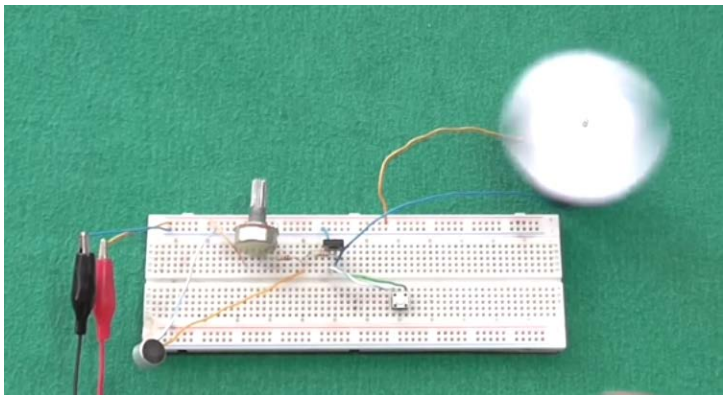
En realidad, un micrófono es una especie de tímpano que nos permite captar los cambios en la presión sonora lo que acentúa más la comparación con el sentido del oído

Micrófono ABM-715-RC Micrófono NXT

Como un ejemplo típico de la utilización de estos sensores seria la realización de un perro guardián, es decir nuestro robot se desplazará cada vez que escuche un sonido.

Tambien podríamos hacer que nuestro robot se desplazará más rápido cuanto mayor es el sonido ambiente.

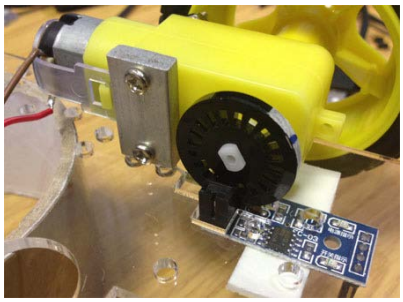
El siguiente enlace nos muestra cómo fabricar un detector de aplauso de manera muy sencilla.

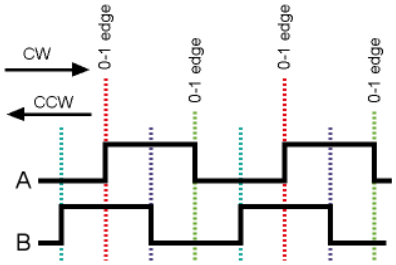


https://www.youtube.com/watch?v=VZXRy8zdw-w

## 3.5. SENSOR DE MOVIMIENTO-ENCODER.

Este sensor nos permite digitalizar el desplazamiento por ejemplo en el caso de una rueda, saber en número de vueltas que ha realizado.
Existen de muchos tipos de generación óptica, magnética…etc.
Si tenemos un solo canal en el sensor solo podremos saber cuento movimiento se ha realizado, pero si introducimos 2 canales podremos saber también el sentido del movimiento.






Un ejemplo de utilizacion del encoder serie el espuesto en el Tema 1 de este curso ya que gracias a el podemos calcular de forma senciallla la distancia recorrida por nuestro robot.

https://youtu.be/E5qvTD9qPXY

## 3.6. SENSOR DE TEMPERATURA.

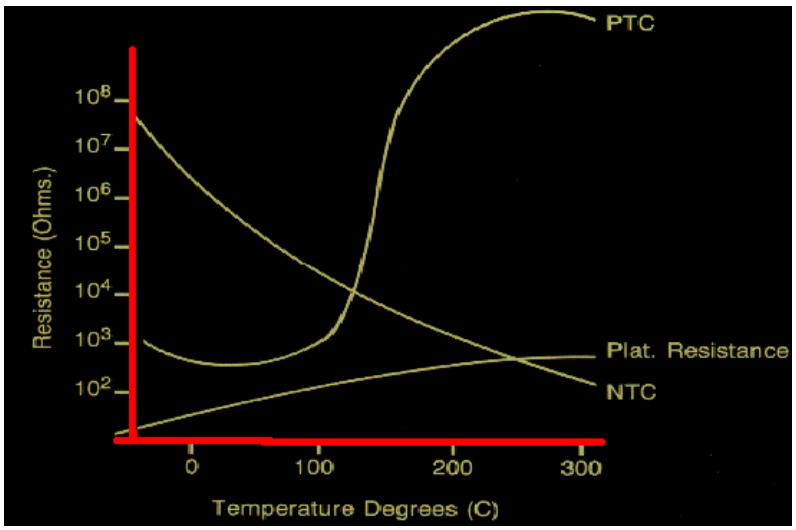
Estos sensores se utilizan para el "sensado" de la temperatura que puede ser ambiente como de cualquier superficie.
Estos sensores aprovechan una característica física de algunos materiales semiconductores estos materiales son capaces de cambiar su resistencia eléctrica en función de la temperatura.

Por lo tanto, aprovechando esta propiedad podemos transformar la temperatura en señales eléctricas.

Existe una gran variedad de comportamiento según el sensor de temperatura que se utilice.

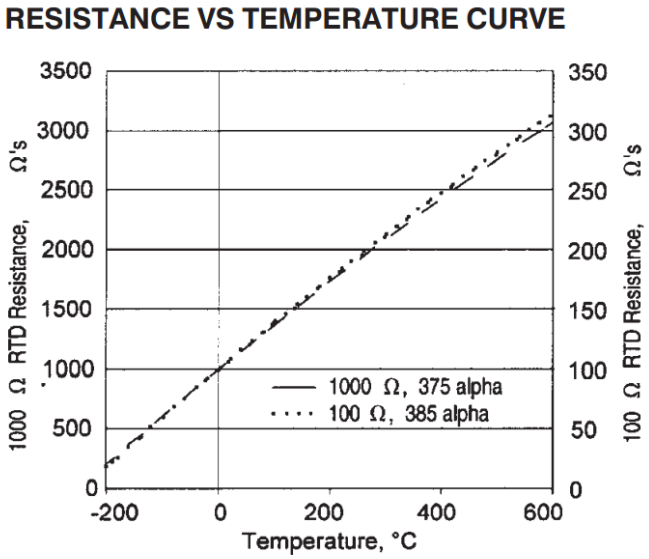
La siguiente lista nos muestra los más utilizados.

- NTC (Coeficiente de temperatura negativo): Estos sensores de temperatura decrecen su resistencia según aumentamos la temperatura.
- PTC (Coeficiente de temperatura positivo): Estos sensores de temperatura trabajan de manera contraria a los NTC ya que aumenta su resistencia con la temperatura.
- Pt100: Este sensor consiste en un alambre de platino que a 0 ºC tiene una resistencia de 100ohm y que al aumentar su temperatura aumenta su resistencia.
- Pt1000: este es análogo al Pt1000, pero a 0ºC se tiene una resistencia de 1000ohm.




Gráfica de Respuesta de sensores NTC y PTC.

Un sensor de temperatura típico en la robótica es el LM35DZ TO92 que es un sensor de temperatura que nos proporciona una salida lineal y cada grado Celsius equivale a 10mv



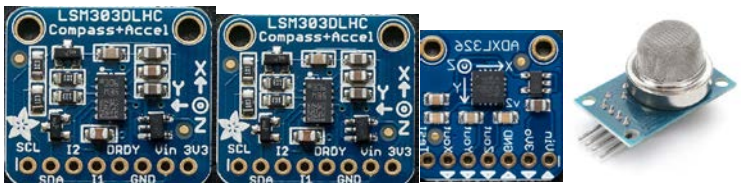

Grafica de la variación de la resistencia del Pt100 y Pt1000 con respecto a la temperatura

Un ejemplo de utilización es este sensor seria el control de la temperatura del agua de una pecera, un terrario o invernadero.

## 3.7. Otros

Existen una gran gama de sensores casi podríamos decir que cada magnitud física tiene su propio sensor, como podemos ver en el siguiente listado.

- Aceleración: Este sensor es capaz de "sensar" la aceleración a la que está sometida y transformarla en una señal eléctrica de baja corriente.
- Giróscopo: Estos sensores son capaces de detectar los cambios de posición y transfórmalos en una señal eléctrica de baja corriente.
- Presión: Estos sensores aprovechan la propiedad de algunos materiales cerámico o cristales iónicos para generar una pequeña energía eléctrica cunado son deformados.
- Humedad: Estos sensores transforman la humedad relativa en señales eléctricas.
- Gases: Estos sensores se encargan del "sensado" de un determinado gas podemos encontrarlos para sensar por ejemplo del oxígeno, Dióxido de carbono o cualquier otro gas.




Distintos sensores.

# TEMA 2.2. ELEMENTOS DE UN ROBOT EDUCATIVO II

## 1. ACTUADORES (SALIDAS).

Podemos definir los actuadores como dispositivos capaces de convertir energía eléctrica en una magnitud física por lo tanto podemos decir que serán los dispositivos electrónicos (en caso de este curso) que podrán interactuar con el mundo exterior.

Como ocurría con los sensores se dispone de un gran abanico de opciones para interactuar con el mundo real.

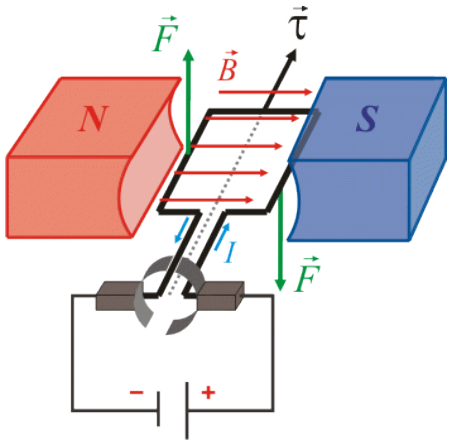
En los siguientes apartados vamos a repasar los actuadores típicos y más utilizados en la Robótica Educativa.

Alicia Asín Pérez. CEO de Libelium.

### 1.1. MOTORES DC.

Estos elementos electro-mecánicos transformaran la energía eléctrica en movimiento.

Los motores DC o también conocidos como motores de corriente continua y suelen ser los más habituales ya que son los más sencillos de controlar y de montar por lo tanto son ideales para la iniciación en la Robótica.




Esquema Motor DC.

Estos motores están compuestos por dos partes **estator** y **rotor.**

- El **estator** es la parte mecánica donde se sitúan los polos del imán.

- El **rotor** es la parte móvil de nuestro motor en la cual tenemos un núcleo y un devanado al que llega la corriente a través de las escobillas.

La manera en la que funciona es la siguiente:

La corriente eléctrica circula por el devanado del **rotor** creando campos electromagnéticos. Este campo interactúa con el campo magnético creado por los imanes del **estator,** generando un rechazo entre los polos del imán del estator y el rotor, creando un par de fuerza, donde el *rotor* girara en un sentido único de forma permanente.

Si se quisiera cambiar el sentido del giro deberemos cambiar el sentido de la corriente en el caso de utilizar baterías o pilas solo tendríamos que invertir la polaridad de estas.

Motor DC con Rueda.

## 1.2. Motor Paso a Paso.

Un motor paso a paso como en el caso anterior es un elemento electro-mecanico que transformara la energía eléctrica en movimiento.

Mientras el motor convencinal gira libremente al aplicarle tensión, el motor paso a paso gira un determinado angulo, es decir nos permite transformar impulsos electricos en movimientos de giro controlados.

Estos desplazamientos angulares pueden ser desde 1.80º hasta 90º.

Estos motores son ideales cuando lo que queremos es un posicinaminto muy exacto y una buena regulacion de velocidad pero al coste de un conexionado y control infinitamente mas complicado que en el caso de los motores DC.

Un ejemplo de utilizacion de estos motores serian las famosas impresoras 3D ya que necesitamos “mucha precision” para relizar las figuras.

https://www.youtube.com/watch?v=dM7j_ULjvc0

Al igual que los motores de DC estos estos se componen del **estator** y del **rotor.**

- El **estator** es la parte fija construida a base de cavidades en las que van depositadas las bobinas.

- El **rotor** parte móvil construida mediante un imán permanente.

Este conjunto va montado sobre un eje soportado por dos cojinetes que le permiten girar libremente.

Motor paso a paso.

El método de funcionamiento es el siguiente:

Al excitar el estator, se crearán los polos N-S, provocando la variación del campo magnético formado.

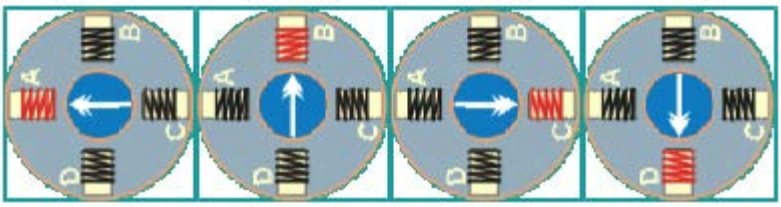
La respuesta del **rotor** será seguir el movimiento de dicho campo (tenderá a buscar la posición de equilibrio magnético), es decir, orientará sus polos NORTE-SUR hacia los polos SUR-NORTE del **estator**, respectivamente.

Cuando el **rotor** alcanza esta posición de equilibrio, el estator cambia la orientación de sus polos y se tratará de buscar la nueva posición de equilibrio.

Manteniendo dicha situación de manera continuada, se conseguirá un movimiento giratorio y continuo del **rotor**, produciéndose de este modo el giro del eje del motor, y a la vez la transformación de una energía eléctrica en otra mecánica en forma de movimiento circular.

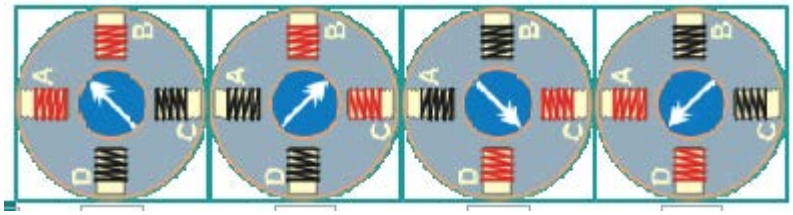
Para controlar estos motores podemos diferenciar 3 grandes métodos según la secuencia de encendido de las bobinas.

- **Paso simple:** consiste en encender cada una de las bobinas de manera secuencial el inconveniente de este método es que no se consigue mucha fuerza ya que solo una bobina sujeta y arrastra el **rotor.**



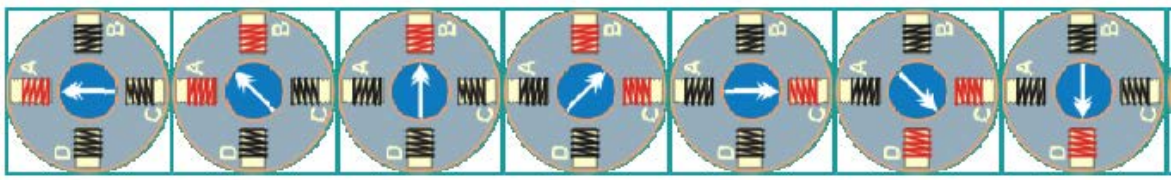

Esquema de funcinamiento paso simple.

- **Paso Doble**: con el paso doble activamos las bobinas de dos en dos consiguiendo un campo magnetico mas elevado por lo tanto los moviemientos son mas bruscos para conseguinmos una mayor potencia.




Esquema de funcionamiento paso doble.

- **Medio Paso**: Es la combinación de los métodos anteriores. Con este método conseguimos mover el motor con pasos más pequeños y precisos, es decir tenemos el doble de pasos en un giro completo del motor.




Esquema de movimiento de Medio paso.

En el siguiente enlace os dejo un video muy interesante de cómo utilizar el motor de un ventilador de Pc, unos imanes y un led para conseguir una linterna de energía gratis.

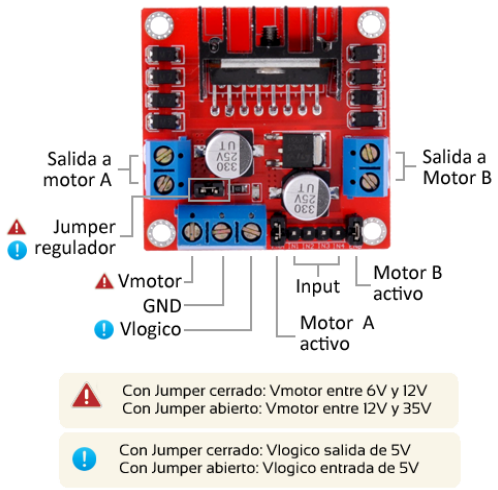
https://www.youtube.com/watch?v=jkJNNpGncZA

## 1.3. DRIVER PARA MOTORES.

Hasta el momento hemos revisado los tipos de motores más usados en la robótica.

En el caso de los motores DC hemos visto que eran controlados por una tensión y que para cambiar el sentido de la rotación tendremos que cambiar la polaridad de la circulación de corriente, pero no hemos dicho cuáles son los niveles de corriente que se necesitan para poder mover nuestro motor y aquí reside el primer problema ya que nuestras unidades de control nos proporcionan unas corrientes de **decenas de miliamperios** ( en caso de Arduino 40mA en el mejor de los casos) y estas no son suficientes para para poder desarrollar el pares óptimos en los motores .

Para solucionar este problema introducimos un dispositivo nuevo que son los drivers de corriente estos sistemas nos permitirán dotar de la corriente suficiente al motor (ya que alimentan el motor directamente desde nuestra fuente de energía) y además podremos cambiar la polaridad de la corriente sin tener que invertir de manera manual el sentido de esta, permitiendo que podemos cambiar el sentido de la rotación mediante las salidas de nuestra unidad de control.

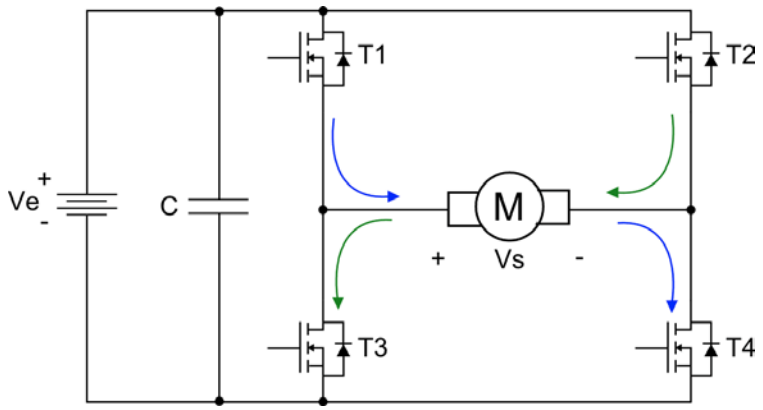



Driver para motores.

Estos sistemas suelen ser puentes en H, pero ¿Qué quiere decir que es un puente en H?

Estos sistemas son cuatro “interruptores” (Transistores) que abriendo unos y cerrando otros fijaran caminos contrarios de circulación de la corriente pudiendo invertir el giro del motor.

La siguiente figura muestra el esquema de cómo están conectados al motor.




Esquema Puente en H.

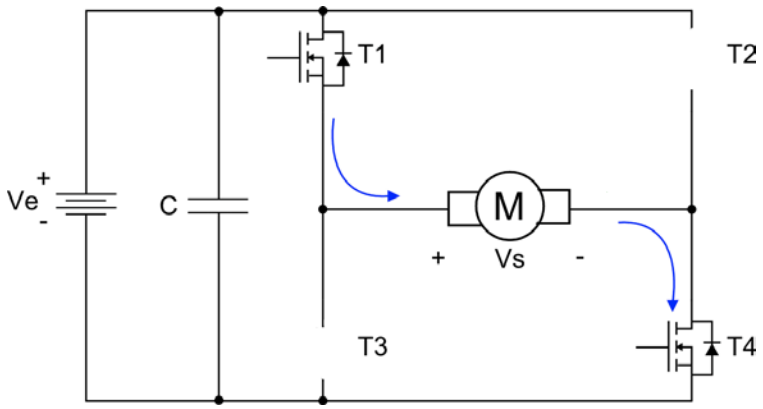
Lo primero que tenemos que se puede observar en el esquema, es que no proporcionamos la corriente al motor mediante la unidad de control, sino que la propia batería o fuente de alimentación de nuestro robot será la que la proporcione.

Con nuestra unidad de control lo que haremos es controlar la apertura y cierre **T1, T2, T3 y T4.**

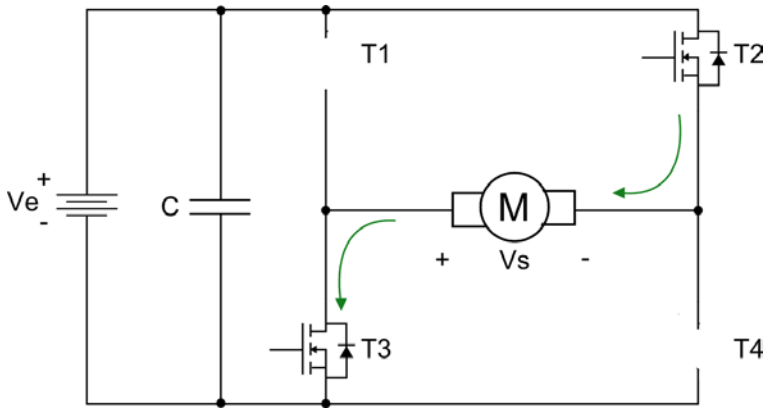
La siguiente tabla nos muestra el funcionamiento del puente H.

| T1 | T2 | T3 | T4 | GIRO |
|---|---|---|---|---|
| ON | OFF | OFF | ON | GIRO SENTIDO 1 |
| OFF | ON | ON | OFF | GIRO SENTIDO 2 |

Las siguientes figuras nos muestran cómo sería el esquema eléctrico para cambiar el sentido de las rotaciones.




Sentido de Giro 1.




Sentido de Giro2.

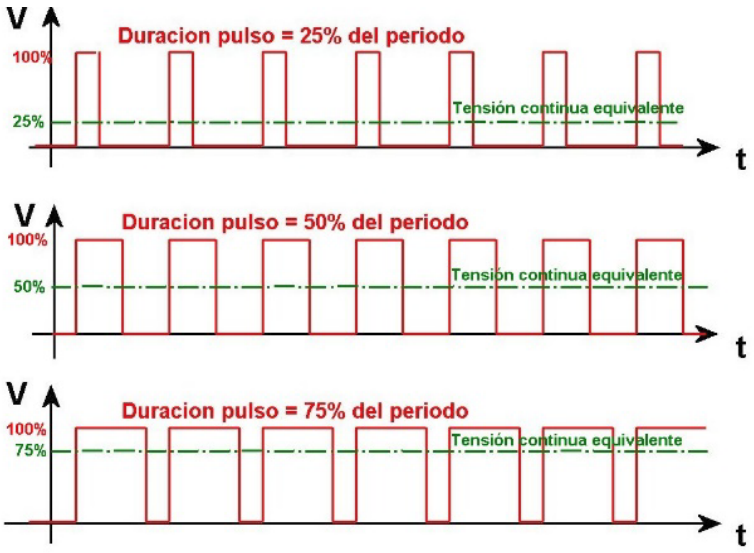
Una vez solucionado la problemática la alimentación y el cambio de sentido se nos presenta la siguiente incógnita ¿Cómo variamos la velocidad?

Pues lo primero que se no acurre es limitar la tensión que suministramos al motor y en consecuencia disminuimos la corriente que circula por el produciendo una disminución de la velocidad.

Ya que no podemos regular la tensión que aplicamos solo con nuestra unidad de control de manera automática para conseguir esto deberemos aplicar una **modulación por ancho de pulso (**PWM).
La **modulación por ancho de pulso** (PWM) consiste en no proporcionar una tensión constante a los interruptores (Transistores) del puente en H que estemos cerrando sino en conmutar el pin, generar pulsos, con esto lo que obtenemos es que tensión en los extremos del motor decrece cuanto menos tiempo este el pulso en estado alto.

- **Nota: Tener en cuenta la unidad de control tiene una tensión de salida constante, es decir que proporciona el nivel de tensión de la alimentación o 0V.**

La siguiente figura muestra cómo funciona la modulación por ancho de pulso.

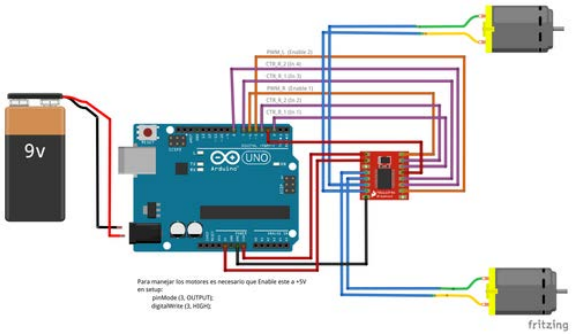



Esquema de PWM.

Como podemos ver si el porcentaje de tiempo que nos mantenemos en estado alto (estamos proporcionado tensión con el pin de la unidad de control) es el porcentaje de tensión continua que aplicamos al motor por lo tanto a menos tiempo en estado alto más lento ira el motor.

**Ejemplo.**

Si tenemos una unidad de control que en sus salidas proporciona 5v y aplicamos un PWM con una duración de pulso del 50% estaríamos aplicando una tensión proporcional de 2,5V al motor, por lo tanto, estaríamos a la mitad de la velocidad máxima que podemos obtener.



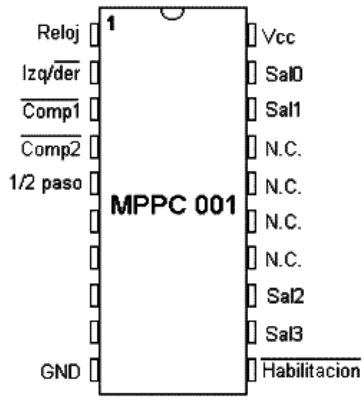

Arduino con driver de motores.

## 1.4. CONTROLADOR DE MOTOR PASO.

Al igual que con los motores DC para los motores paso a paso tenemos unos controladores que se encargan de suministrar la corriente necesaria, controlar el sentido del giro, la velocidad y además en caso de los motores pasos a paso nos permitirán controlar el **paso** con el que moveremos el motor.

Otra ventaja es que estos controladores necesitan pocas salidas de nuestra unidad de control para realizar el movimiento deseado, ya que si recordamos los motores paso a paso tenían el inconveniente de necesitar muchas líneas de control y por lo tanto simplificaremos la programación en la unidad de control.

Si tomamos el ejemplo del integrado **MPPC 001** solo necesitaremos 3 salidas de nuestra unidad de control para controlar el movimiento, 2 controlaran el sentido del giro y otra para fijar el Paso (paso simple y medio paso).




Esquema controlador MPPC 001.

En el siguiente enlace podremos ver como conectan y controlan un motor paso a paso mediante Arduino y el controlador **ULN2003**.

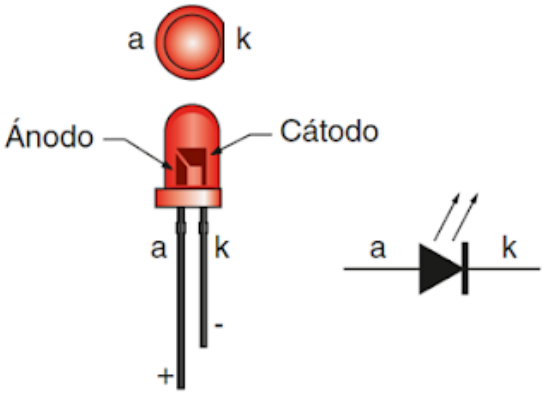
https://www.youtube.com/watch?v=2-nVV9S7leM

Controlador ULN2003.

## 1.5. LED.

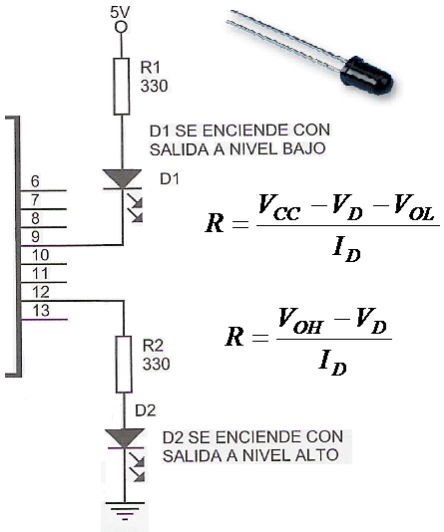
Los dispositivos emisores de luz (LED) lo podemos definir como unas pequeñas bombillas que utilizaremos como indicadores en nuestros robots.

Los LED son diodos y por lo tanto tendremos un ánodo y un cátodo su funcionamiento es muy sencillo cuando la tensión en ánodo es suficientemente superior a la tensión en el cátodo estos se iluminan.




Esquema y símbolo del diodo LED.

La siguiente figura nos muestra como conectar un diodo led a nuestra unidad de control.




Esquema de implementación en una unidad de control.

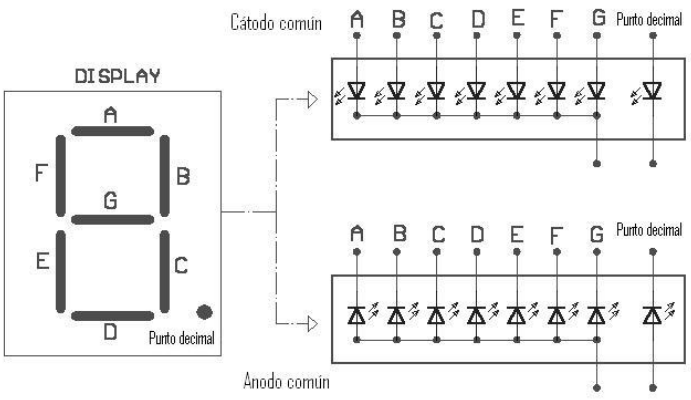
Como podemos ver en el esquema podemos activar el LED tanto poniendo 0V a la salida como 5V.
También tenemos que ver que en el esquema se pone una resistencia en serie con el diodo LED esto es debido a que debemos limitar la corriente que circula por el para evitar que se deteriore por ejemplo ya que una sobre-corriente sobre el acabaría destruirlo.

El siguiente video muestra cómo realizar un cubo de LED 4x4x4.

https://www.youtube.com/watch?v=TesnKjba0l0

## 1.6. DISPLAY DE 7 SEGMENTOS.

Estos dispositivos podríamos decir que son carteles luminosos que están formados por 7 u 8 LED (si muestran el punto decimal) los cuales nos permiten representar números, letras…etc.




Esquema Display de 7 segmentos con punto decimal.

Tenemos dos tipos según su conexionado.

- Cátodo Común: Tiene unidos todos los cátodos de los diodos internos.
- Ánodo Común: Tiene unidos todos los ánodos de los diodos internos.

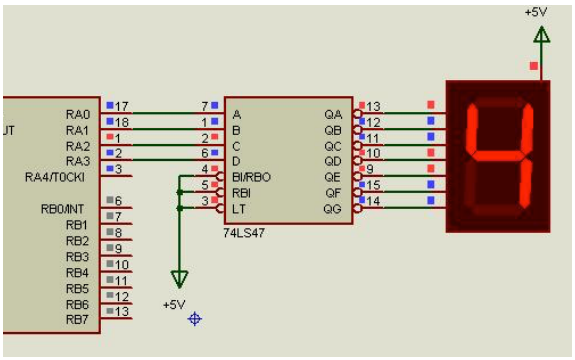
Es importante saber qué tipo de display tenemos ya que como comentamos en el punto anterior necesitamos tener una tensión lo suficientemente superior en el ánodo respecto cátodo para iluminar el led.

Como podemos apreciar para controlar el Display necesitaremos 8 líneas de control de nuestra unidad de control por lo tanto siempre es aconsejable utilizar conversores de BCD a Display de 7 segmentos ya que así podemos controlar este dispositivo solo con cuatro líneas de control.

El siguiente video nos introducirá en la utilización del código BCD.

https://www.youtube.com/watch?v=NgVKDkQlwU0

Un conversor típico es el **74LS47** y en la siguiente figura veremos el esquema de conexión de la unidad de control, conversor y Display de 7 segmentos.




Conexión de conversor.

La siguiente tabla nos muestra cómo cambian las salidas del **74lS47** con respecto a la entrada.

**TRUTH TABLE**

| | INPUTS | | | | | | | OUTPUTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DECIMAL OR FUNCTION | $\overline{LT}$ | $\overline{RBI}$ | D | C | B | A | $\overline{BI/RBO}$ | $\overline{a}$ | $\overline{b}$ | $\overline{c}$ | $\overline{d}$ | $\overline{e}$ | f | $\overline{g}$ | NOTE |
| 0 | H | H | L | L | L | L | H | L | L | L | L | L | L | H | A |
| 1 | H | X | L | L | L | H | H | H | L | L | H | H | H | H | A |
| 2 | H | X | L | L | H | L | H | L | L | H | L | L | H | L | |
| 3 | H | X | L | L | H | H | H | L | L | L | L | H | H | L | |
| 4 | H | X | L | H | L | L | H | H | L | L | H | H | L | L | |
| 5 | H | X | L | H | L | H | H | L | H | L | L | H | L | L | |
| 6 | H | X | L | H | H | L | H | H | H | L | L | L | L | L | |
| 7 | H | X | L | H | H | H | H | L | L | L | H | H | H | H | |
| 8 | H | X | H | L | L | L | H | L | L | L | L | L | L | L | |
| 9 | H | X | H | L | L | H | H | L | L | L | H | H | L | L | |
| 10 | H | X | H | L | H | L | H | H | H | H | L | L | H | L | |
| 11 | H | X | H | L | H | H | H | H | H | L | L | H | H | L | |
| 12 | H | X | H | H | L | L | H | H | L | H | H | H | L | L | |
| 13 | H | X | H | H | L | H | H | L | H | H | L | H | L | L | |
| 14 | H | X | H | H | H | L | H | H | H | H | L | L | L | L | |
| 15 | H | X | H | H | H | H | H | H | H | H | H | H | H | H | |
| BI | X | X | X | X | X | X | L | H | H | H | H | H | H | H | B |
| RBI | H | L | L | L | L | L | L | H | H | H | H | H | H | H | C |
| LT | L | X | X | X | X | X | H | L | L | L | L | L | L | L | D |

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

Tabla de verdad 74LS47

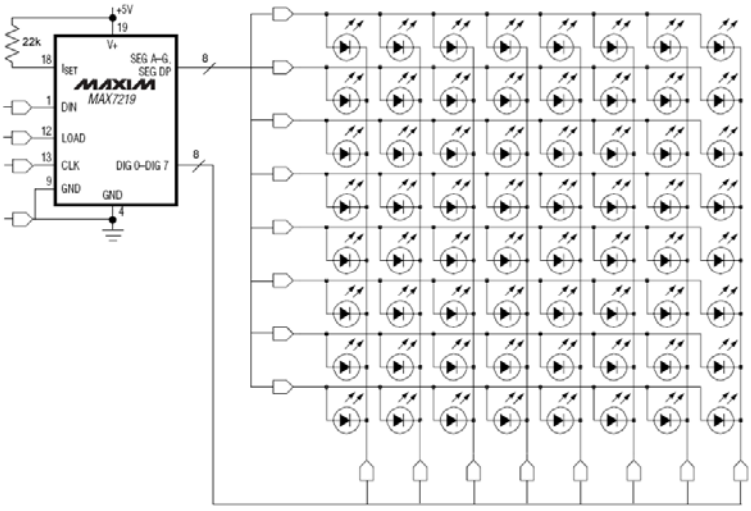
## 1.7. DISPLAY GRAFICO DE MATRIZ LED.

Como su propio nombre describe el Display Grafico de Matriz LED es un matriz de LED la cual tendrá una dimensión NxM.
Ya que necesitaríamos una cantidad de líneas de control demasiado grande, al igual que en caso del Display de 7 Segmentos, utilizaremos un driver que nos permitirá elegir en la posición del LED a encender.

Matriz de LED.

Para poder hacer gráficos **el procedimiento consiste en hacer un barrido por filas** (o columnas). Encendemos todos los LED de una única fila (o columna), y a continuación cambiamos a la siguiente. Sólo una fila (o columna) está encendida cada instante, pero al hacerlo rápidamente, el efecto en nuestra visión es ver toda la imagen formada de forma simultánea.
Este efecto se denomina **"Persistencia de visión" (POV)**, y es consecuencia de la forma en la que los humanos percibimos el movimiento. Es muy empleado en electrónica e informática.

La utilización de estos drivers como en caso del **MAX7219** nos permite controlar la matriz comunicándonos con este y solo con 3 líneas (en este caso comunicación **SPI**) podremos representa lo que queramos en la matriz.




Esquema de driver para matriz de LED.

## 1.8.LCD.

Estos dispositivos son la evolución de la matriz de led, los LCD, (representación visual por cristal líquido) son pequeñas pantallas formadas por una gran cantidad de pixeles por lo tanto al igual que en el caso anterior tendrá un driver que permitirá controlas cada una de estos pixeles.

Por lo tanto, pasara igual que en caso anterior utilizaremos un protocolo de comunicación, que informara al display LCD que posiciones encender.

También tenemos que decir que al contrario que los dispositivos anteriores el driver ya va integrado en la pantalla por lo que no necesitamos un conexionado extra.


Display LCD.

## 1.9.ZUMBADORES.

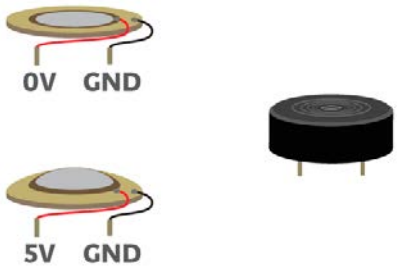
Estos dispositivos son capaces de convertir señales eléctricas en señales acústicas.

En caso de los zumbadores los podemos clasificar en 3 tipos.

- Piezoeléctricos.
- Electromagnéticos.
- Con/Sin oscilador.

Su funcionamiento es sencillo emitimos una señal eléctrica con una frecuencia determinada y este dispositivo lo transformara en el tono correspondiente.

Por ejemplo, si queremos reproducir un ***La 3*** tendríamos que generar una señal de 440 Hz de frecuencia.




Zumbador.

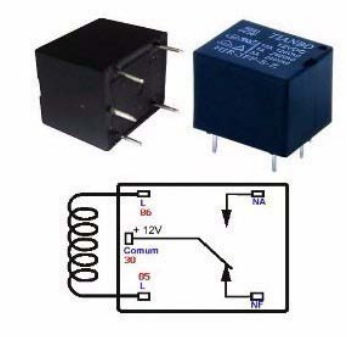
## 1.10. RELES.

Estos actuadores los podemos definir como interruptores que pueden ser controlados por las unidades de control.

Gracias a ellos podemos bloquear o activar por ejemplo un sensor u otro actuador.

Los podemos clasificar en 3 tipos.

- Electromagnéticos.
- Reed.
- Estado sólido.




Relés Electromagnéticos.

El siguiente video nos introducirá en la utilización de relés con Arduino.

https://www.youtube.com/watch?v=J_JF-KTuClI

# 2. PLATAFORMAS MECÁNICAS.

Hasta el momento hemos revisado la unidad de control, sensores y actuadores, pero no hemos visto la unión de todas sus partes para formar un robot.

Cuando nos referimos a la plataforma mecánica queremos decir la estructura que conformara nuestro robot. Es la estructura no es una fija, ya que se tendrá que adecuar a la tarea que tenga que realizar nuestro robot.

Normalmente cuando pensamos en robots nos viene a la mente una forma humanoide, pero esto se aleja mucho de la realidad ya que por ejemplo un brazo robot es una estructura mucho más extendida que la forma humanoide.

Diversas plataformas mecánicas.

La plataforma mecánica más extendida y utilizada en la robótica educativa es la que denominaremos tribot.

Esta estructura está formada por una plancha perforada en la cual instalaremos dos motores, la unidad de control y los sensores que queramos utilizar.

Tambien tenemos una rueda muerta (denominada así porque no tiene motor) para facilitar el giro.

Plataforma mBot.

.

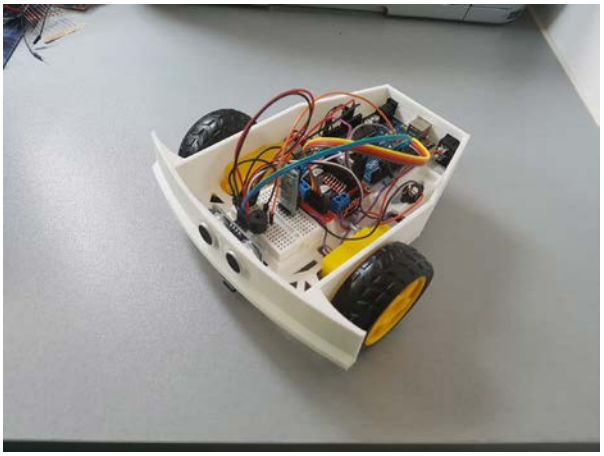
Una cuestión importante a tener en cuenta en estas plataformas mecánicas es que no tenemos un volante para proporcionar el giro de la plataforma.

La manera de hacer que nuestro robot gire en un sentido o en otro será bloquear una rueda y avanzar con la otra, por lo que el control de los giros conlleva una experimentación con la plataforma.

**Nota: Si queremos que los giros sean más rápidos y cerrados también podemos invertir el sentido de una rueda en vez de bloquearla.**

Un error muy común en el montaje de nuestro robot es la colocación de los sensores por lo que necesitamos tener en cuenta que mide el sensor y cuál es la posición más óptima para colocarlo en nuestra plataforma, ya que la mala colocación de estos puede influir en su lectura o incluso entorpecer la labor de otros sensores y actuadores.

Por lo tanto, podemos decir que es tan importante, en contra de lo que suele opinar la gente, la mecánica de nuestro robot como la programación.



Plataforma tribot.

# TEMA 3. REVISIÓN DE ROBOTS EDUCATIVOS

## 1. INTRODUCCIÓN

En la actualidad podemos encontrar una gran variedad de Robots Educativos en diferentes tiendas on-line dedicadas exclusivamente a la venta de los mismos, en distribuidores de material educativo en tiendas de electrónica o, incluso, en grandes almacenes o tiendas de electrónica doméstica.





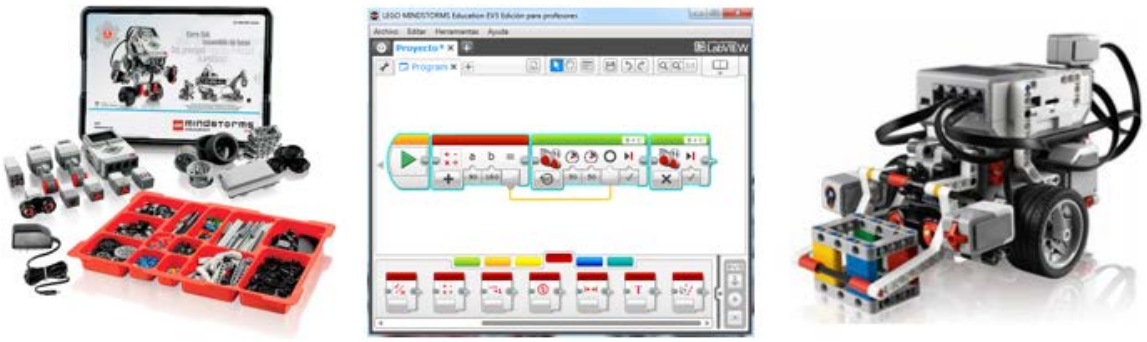


Para realizar la elección correcta del robot a emplear, es necesario tener en cuenta una serie de factores centrados en el nivel educativo de los alumnos en el que se va a utilizar el robot y la infraestructura que se dispone en el centro educativo. En este sentido, de forma más concreta en este tema se van a comparar diferentes plataformas educativas teniendo en cuenta los siguientes ítems:

- Etapa educativa recomendada.
- Sistemas operativos con los que puede ser programado (si es compatible con Linux).
- Precio del robot.
- Sensores y actuadores que dispone.
- Posibilidad de realizar diferentes montajes.
- Robustez.
- Sencillez de conexionado de los diferentes módulos.
- Entorno de programación por bloques.
- Recursos didácticos que proporciona.

## 2. Robot Mindstorms EV3 de LEGO Education

El robot EV3 representa una plataforma muy completa para poder llevar a cabo una gran multitud de proyectos. Además de contar con un elevado número de engranajes y piezas que permiten realizar infinitos modelos de robots, es una plataforma muy robusta frente a golpes y duración de batería. Otra de sus ventajas es la plataforma de programación que proporciona, ya que es muy intuitiva y cuenta con guías de montaje y programación paso a paso de diferentes modelos, así como, guías didácticas para tratar las materias STEM.



A pesar de todas estas ventajas, es uno de los robots educativos más caros del mercado y no es compatible con el sistema operativo Linux, por lo que es complica su uso en la mayoría de centros educativos. En algunos centros, debido a su coste, emplean un único robot por clase y un único PC con sistema operativo Windows para poder programarlo, organizando a los alumnos por grupos de trabajo. De este modo, se asignan diferentes tareas que van realizando los grupos de forma rotativa: búsqueda de información sobre la aplicación a realizar y programar en el robot, montaje del robot, programación del mismo y documentación del proyecto.

La siguiente tabla resumen las características de dicho robot:

| | |
|---|---|
| **Etapa educativa** | Secundaria y último ciclo de Primaria |
| **Compatibilidad con Linux** | No |
| **Precio** | 492 € (Kit Educativo) |
| **Sensores y actuadores** | Giróscopo, Ultrasónico, Color/Luz, Contacto, 3 motores |

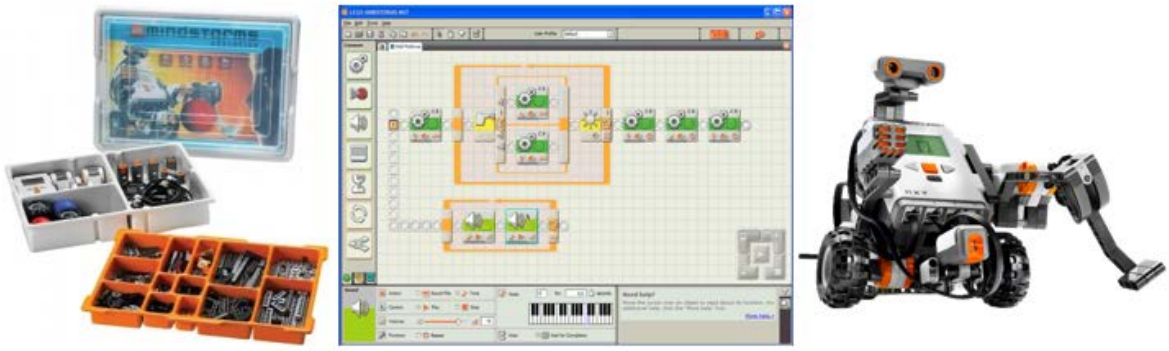
| | |
|---|---|
| **Diferentes montajes** | Sí, 541 piezas de construcción |
| **Robustez** | Muy robusto |
| **Conexionado** | Muy sencillo |
| **Entorno de programación** | Amigable y sencillo. Entorno para adquirir datos y representar gráficas |
| **Recursos didácticos** | Gran cantidad de programaciones de aula y guías de montaje y programación |

En el siguiente vídeo se muestra el robot educativo que propone LEGO para empezar a trabajar con los diferentes motores y sensores. También se explica la parte de adquisición de datos que integra la plataforma para realizar la medida de diferentes magnitudes empleando los sensores:


https://youtu.be/9SmEv8xx64w

## 3. Robot Mindstorms NXT de LEGO Education

El robot NXT de LEGO es la versión que precede al modelo EV3. Este robot es muy presenta características muy similares al EV3, pero actualmente se encuentra descatalogado. Hemos querido incluirlo porque algunos centros educativos cuentan con unidades de estos robots y, a pesar de su descatalogación, todavía existen plataformas en el que se encuentra disponible para su adquisición.



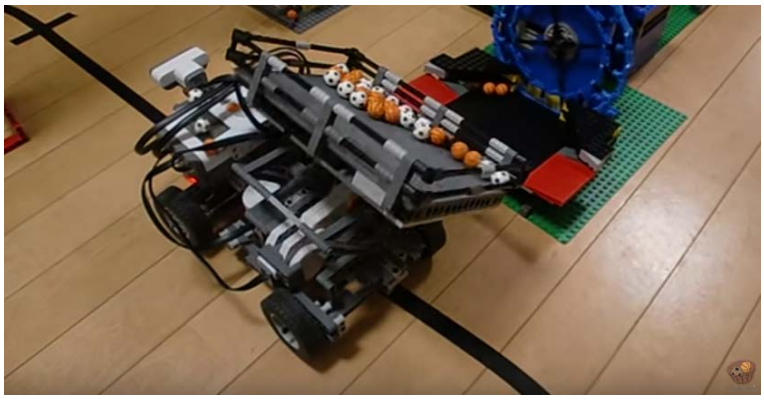
En cuanto a características de elementos mecánicos, sensores y actuadores, es muy similar a EV3, pero quizá su ventaja frente a su actualización radica en la mayor sencillez de su entorno de programación. La versión actual del EV3 posee un número de recursos mayor pero no es tan sencilla como la del NXT (NXT Programming 2.1).

La siguiente tabla resumen las características de dicho robot:

| | |
|---|---|
| **Etapa educativa** | Secundaria y Primaria (a partir de 3º) |
| **Compatibilidad con Linux** | No |
| **Precio** | 390 € (Kit Educativo) |
| **Sensores y actuadores** | Sonido, Ultrasónico, Luz, Contacto, 3 motores |
| **Diferentes montajes** | Sí, 470 piezas de construcción |
| **Robustez** | Muy robusto |
| **Conexionado** | Muy sencillo |
| **Entorno de programación** | Amigable y sencillo. Entorno para adquirir datos y representar gráficas |

| | |
|---|---|
| **Recursos didácticos** | Gran cantidad de guías de montaje y programación |

En el siguiente vídeo se muestra una línea de producción implementada a base de kits NXT. Este vídeo puede proporcionar ideas para llevar a cabo proyectos colaborativos en los que varios robots interactúen para llevar a cabo una función concreta:



https://youtu.be/rWd3vgLaA_M

## 4. ROBOT TXT ADVANCED FISCHERTECHNIK EDUCATION

El robot TXT Advanced de Fischertechnik ofrece unas características que se asemejan bastante a las propuestas de LEGO. A pesar de no proporcionar tantos recursos didácticos ni la posibilidad de realizar tantos montajes como el EV3 o NXT, la ventaja principal frente a estos es la posibilidad de programarlos con Scratch y, por tanto, con un ordenador basado en Linux. En cuanto al precio, presenta un precio similar al de LEGO. En el siguiente link se indican las instrucciones de instalación del mismo y los bloques que permite emplear para programarlo:

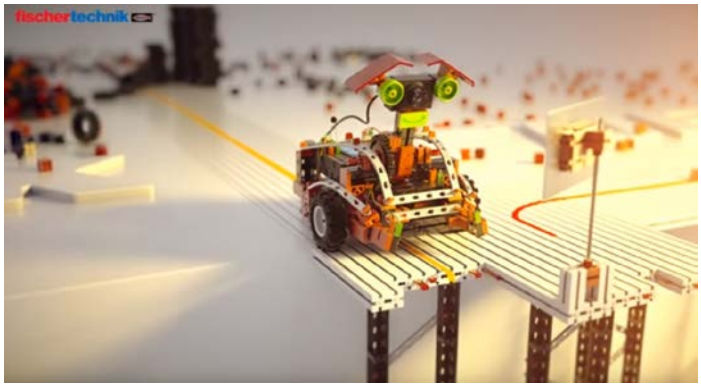
https://ftscratch.github.io/ROBO-TXT/www/es/index.html


La siguiente tabla resumen las características de dicho robot:

| | |
|---|---|
| **Etapa educativa** | Secundaria (debido al manejo de sus sensores) |
| **Compatibilidad con Linux** | Sí, puede programarse a través de un módulo de Scratch |
| **Precio** | 390 € (Kit Educativo) |
| **Sensores y actuadores** | Cámara USB, Pulsadores, Fototransistor, Temperatura, LEDs y 3 motores. |

| | |
|---|---|
| **Diferentes montajes** | Sí, 310 piezas de construcción |
| **Robustez** | Robusto, aunque menos que las plataformas de LEGO. |
| **Conexionado** | Sencillo |
| **Entorno de programación** | Amigable y sencillo (Scratch) |
| **Recursos didácticos** | Reducidos |

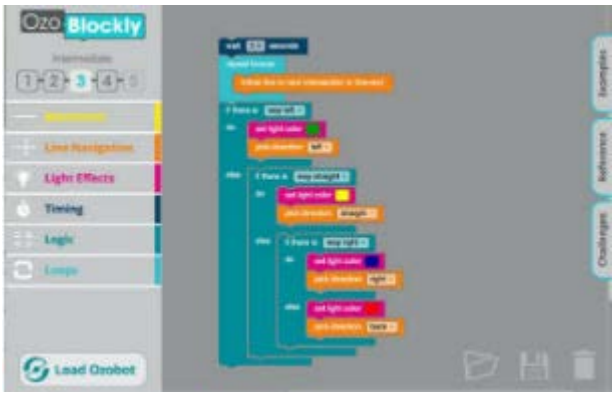
En el siguiente vídeo se puede observar la estructura del robot TXT Advanced:




https://youtu.be/9ESg4O0-QAs

## 5. Robot Ozobot Evo Education

Este es un robot distinto al resto que se van a mostrar en este tema, pero que está teniendo gran éxito para potenciar actividades en las que se trabaja el pensamiento computacional debido a su sencillez y agilidad. De este modo, no es necesario realizar ningún interconexionado ni montaje, ya que se trata de un robot compacto con todos sus componentes integrados. Por tanto, se trata de un robot para trabajar de forma exclusiva la parte de programación mediante una APP que puede instalarse en dispositivos móviles o tabletas con una versión de Android superior a la 4.0, que dispongan conectividad Bluetooth (4.0 o superior) y al menos 2.0 GB de memoria RAM. La programación se realiza mediante bloques con un lenguaje muy similar a Scratch.




En cuanto a los sensores y actuadores del robot, consta de un sensor de proximidad para detectar obstáculos, un sensor óptico (sensor de luz/color) para detectar líneas y códigos de colores, LED de colores y un altavoz.

La siguiente tabla resumen las características de dicho robot:

| | |
|---|---|
| **Etapa educativa** | Secundaria |
| **Compatibilidad con Linux** | No, pero sí con Android 4.0 o superior |
| **Precio** | 120 € (Kit Educativo) |
| **Sensores y actuadores** | Sensor de proximidad, sensor de luz/color, LEDs de colores, un altavoz y dos motores. |
| **Diferentes montajes** | No |
| **Robustez** | Muy robusto, no es modular. |
| **Conexionado** | No existe interconexionado. |

| | |
|---|---|
| **Entorno de programación** | Amigable y sencillo (parecido a Scratch) |
| **Recursos didácticos** | Moderados |

En el siguiente vídeo se muestran algunas de las características del robot Ozobot:


https://youtu.be/UlZHh_1EqfU

## 6. mBOT ROBOT

El robot mBot representa una de las opciones más competitivas y más realistas para ser empleada en centros educativos, ya que su plataforma mBlock es compatible con Linux e incluso están desarrollando una interfaz web para poder programarlo a través de un navegador como Google Chrome.

En cuanto a las características de la plataforma, posee una gran variedad de sensores que pueden ser intercambiados y se pueden identificar con facilidad dentro de su estructura. La mecánica es uno de los aspectos que lo limitan, puesto que no es la opción más versátil para realizar cambios en su montaje ya que es necesario emplear otro kit.

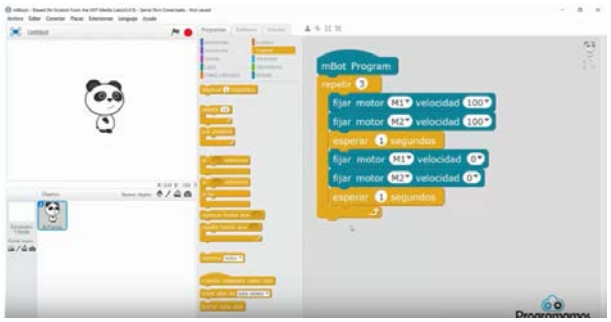

La siguiente tabla resumen las características de dicho robot:

| | |
|---|---|
| **Etapa educativa** | Secundaria y último ciclo de Primaria |
| **Compatibilidad con Linux** | Sí |
| **Precio** | 90 € (Kit más económico ) |
| **Sensores y actuadores** | Sensor de luz, Emisor y receptor de infrarrojos, dos LED RGB, Zumbador, Pulsador, Sensor ultrasonidos, Sensor sigue-líneas, Motores. |
| **Diferentes montajes** | No |
| **Robustez** | Muy robusto, no es modular. |
| **Conexionado** | Muy sencillo y robusto. |

| | |
|---|---|
| **Entorno de programación** | Amigable y sencillo (mBlock, parecido a Scratch) |
| **Recursos didácticos** | Moderados |

En el siguiente vídeo se muestran algunas de las características del robot mBot:

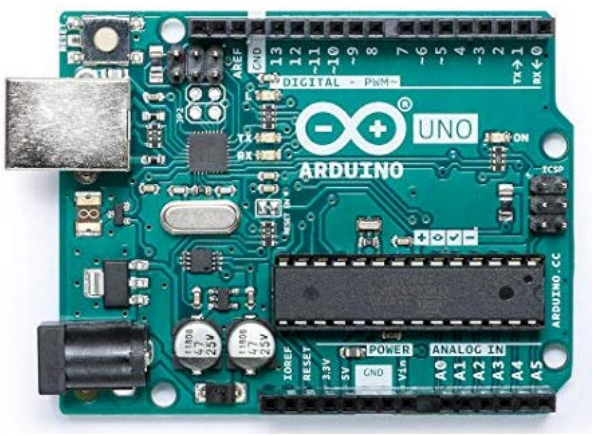

https://youtu.be/chkpweFx6G4




https://youtu.be/EHOyL2hd0I8

## 7. ROBOT DIY BASADO EN ARDUINO

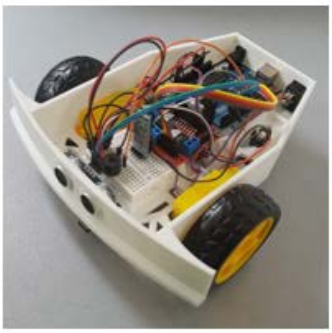
Arduino es una tecnología basada en hardware (parte física de la electrónica) y en software (códigos que se programan y plataforma de programación) libre o también conocida como open access.




En este sentido se identifica Arduino con placas de desarrollo que integran un microcontrolador que permite programar diferentes comportamientos a través de un ordenador. De este modo, a través de sus puertos de conexión (tiras de pines), es posible conectar diferentes sensores y actuadores a la placa de Arduino con el fin de conseguir implementar el robot deseado.


La principal ventaja de Arduino es la posibilidad de desarrollar el robot que se desee y que más se ajuste a las características que se necesitan para llevar a cabo una tarea determinada. Por el contrario, necesita un nivel de conocimientos más elevados que las plataformas anteriormente presentadas debido a que se debe diseñar y crear la estructura del robot,

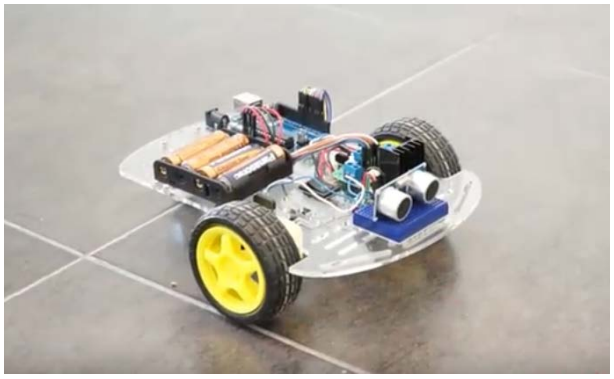
conectar los sensores que se deseen emplear mediante cables, y programar el comportamiento del robot mediante código. No obstante, pueden emplearse otras plataformas de programación, con el fin de facilitar la tarea de programación, por lo que es posible utilizar plataformas como mBlock o Scratch.




La siguiente tabla resumen las características de dicho robot:

| | |
|---|---|
| **Etapa educativa** | Secundaria |
| **Compatibilidad con Linux** | Sí (puede programarse con mBlock) |
| **Precio** | Muy económico, la placa en torno a 10€, un robot básico alrededor de 40€ |
| **Sensores y actuadores** | Una infinita variedad de sensores: ultrasonidos, temperatura, luz, color, humedad, infrarrojos, sonido…motores, LEDs, altavoces, zumbadores.... |
| **Diferentes montajes** | Sí, pero la estructura es DIY, se puede realizar con madera, impresora 3D, bases estándar… |
| **Robustez** | Poco robusto, debido a la interconexión de los módulos por cables, y al bajo coste de los módulos utilizados. |
| **Conexionado** | Muy complejo, cableando cada módulo con la placa Arduino |
| **Entorno de programación** | Puede programarse con lenguaje por código o empleando bloques (Scratch, mBlock, bitBloc…) |
| **Recursos didácticos** | Una gran cantidad de recursos en internet, Youtube, libros especializados… |

En el siguiente vídeo se puede observar el montaje de un robot esquivador de obstáculos basado en Arduino:



https://youtu.be/60kLBEZSf3Q

# TEMA 4: PLATAFORMAS PARA PROGRAMAR ROBOTS EDUCATIVOS

En este tema vamos a revisar las distintas plataformas para la programación de nuestros robots educativos descubriendo sus principales características.

Decir que hoy en día existen infinidad de plataformas, tanto como distintos robots disponibles en el mercado, pero en este tema solo vamos a revisar las más populares y extendidas lo cual nos dará una idea bastante aproximada de lo que podemos encontrar en el mercado y lo que nos pueden aportar.

## 1. IDE DE ARDUINO.

Aunque Arduino IDE nos es aconsejable para la iniciación a la robótica, sobre todos en cursos de primaria y principios de secundaria, si realizaremos una revisión de esta plataforma de programación, ya que muchas de la plataforma que vamos a ver están basadas en este IDE.

El IDE de Arduino corresponde a las siglas de *Integrated Develomnet Envionment* o entorno de desarrollo integrado.

Este entorno trabaja con fichero denominados sketch que será nuestro proyecto principal donde nos permitirá escribir el código.


Cuando guardamos nuestros proyectos el IDE de Arduino creara un archivo *.INO el cual podremos abrir para modificar en cualquier momento.

El lenguaje de programación de Arduino está basado en C++ y es un lenguaje “escrito” y por lo tanto presenta una gran problemática, ya que los lenguajes escritos es necesario conocer la sintaxis del lenguaje, palabras reservadas…etc.

La utilización de estos lenguajes (en este caso sería un lenguaje de nivel medio), nos apartan del objetivo de trabajar el pensamiento computacional, ya que se necesita cierta practica y conocimientos para dominar el lenguaje y en edades tempranas tener que escribir código hace que el alumnado pierda la motivación.

Una gran ventaja del Arduino IDE es la gran cantidad de librerías disponibles ya que posee una gran comunidad de usuarios que nos proporcionan código libre que nos puede facilitar el control de muchos de los sensores y actuadores.

También tenemos que notar que tener una gran cantidad de librerías y ejemplos disponibles es una gran desventaja a la hora de que los alumnos aprendan, ya que en muchas ocasiones los alumnos se dedican a cortar y pegar código, en ocasiones código sin ninguna garantía, con lo que se pierde el fundamento del pensamiento computacional.

El en siguiente tabla vamos a resumir las principales características de Arduino IDE.

| | |
|---|---|
| **Plataformas permitidas** | Linux , Windows y Mac OS. |
| **S.O Lliurex** | Permite la instalación y la programación en Lliurex. |
| **Precio** | No tiene coste. |
| **Programación** | Código basado en lenguaje C++, por líneas de texto y permite el trabajo autónomo del robot. |
| **Curva de aprendizaje** | Medio. |
| **Librerías** | Sin coste y libre distribución.. |

En el siguiente enlace tenéis la página oficial de para la descarga del Arduino IDE.

https://www.arduino.cc/en/Main/Software

El siguiente ejemplo nos mostrara un ejemplo de código mediante el cual podremos sensar la distancia mediante un sensor de ultrasonidos y enviarla al PC mediante una comunicación puerto serie.

```
const int EchoPin = 5;
const int TriggerPin = 6;

void setup() {
   Serial.begin(9600);
   pinMode(TriggerPin, OUTPUT);
   pinMode(EchoPin, INPUT);
}

void loop() {
   int cm = ping(TriggerPin, EchoPin);
   Serial.print("Distancia: ");
   Serial.println(cm);
   delay(1000);
}

int ping(int TriggerPin, int EchoPin) {
   long duration, distanceCm;

   digitalWrite(TriggerPin, LOW);  //para generar un pulso limpio ponemos a LOW 4us
   delayMicroseconds(4);
   digitalWrite(TriggerPin, HIGH);  //generamos Trigger (disparo) de 10us
   delayMicroseconds(10);
   digitalWrite(TriggerPin, LOW);

   duration = pulseIn(EchoPin, HIGH);  //medimos el tiempo entre pulsos, en microsegundos

   distanceCm = duration * 10 / 292/ 2;   //convertimos a distancia, en cm
   return distanceCm;
}
```

En el siguiente ejemplo realizaremos el mismo programa, pero utilizando una librería creada para el uso del sensor de ultrasonidos, la librería NewPing.

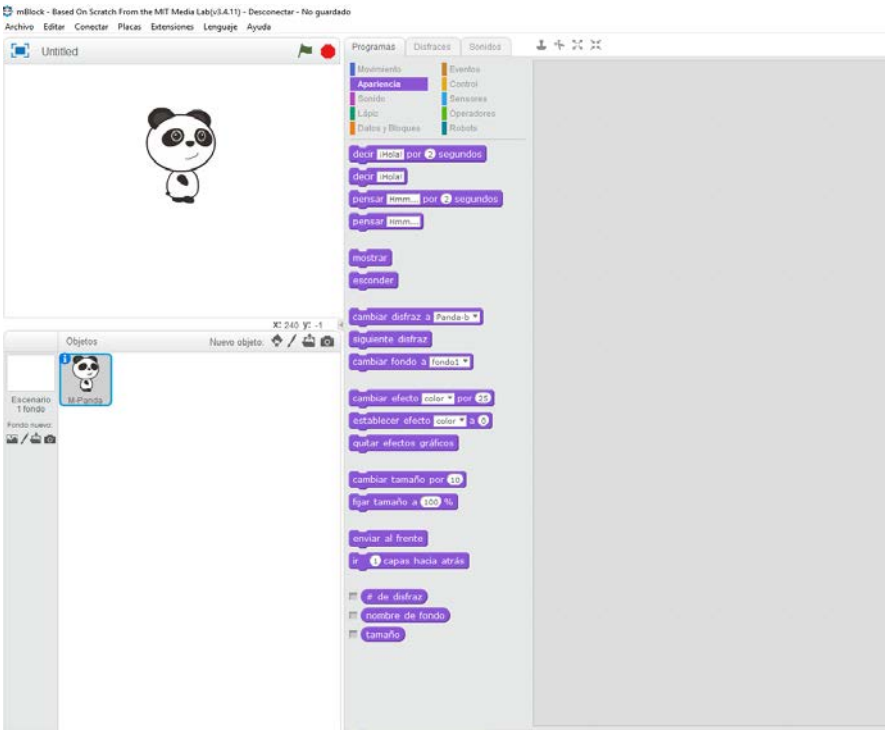
```
#include <NewPing.h>

const int UltrasonicPin = 5;
const int MaxDistance = 200;

NewPing sonar(UltrasonicPin, UltrasonicPin, MaxDistance);

void setup() {
  Serial.begin(9600);
}

void loop() {
  delay(50);                      // esperar 50ms entre pings (29ms como minimo)
  Serial.print(sonar.ping_cm()); // obtener el valor en cm (0 = fuera de rango)
  Serial.println("cm");
}
```

Como podemos ver el uso de librerías simplifica la utilización de sensores, pero tenemos que recordar que uso de estas abstrae la programación a niveles más elevados con lo que se puede perder la compresión de cómo trabaja el sensor y lo necesario para su utilización ya que pasa a ser un componente abstracto para el usuario.

# 2. MBLOCK.

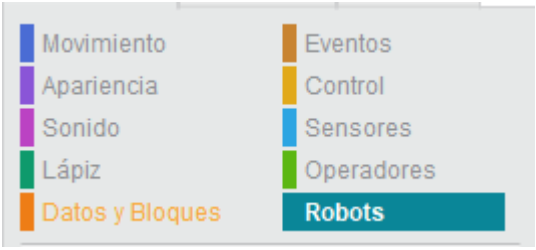
El entorno de programación de mBlock se creó con el objetivo de acercar la programación de robots a los niños, por lo tanto, se eliminará la utilización de código por bloques de instrucciones que se encajaran como piezas de puzles para crear nuestros programas.

El entorno mBlock nace de una adaptación del famoso Scratch, lenguaje que se creó para la creación de videojuegos en el PC, para la programación de robots, aunque tenemos que notar que también se puede utilizar para la creación de proyectos caseros, domótica …etc.

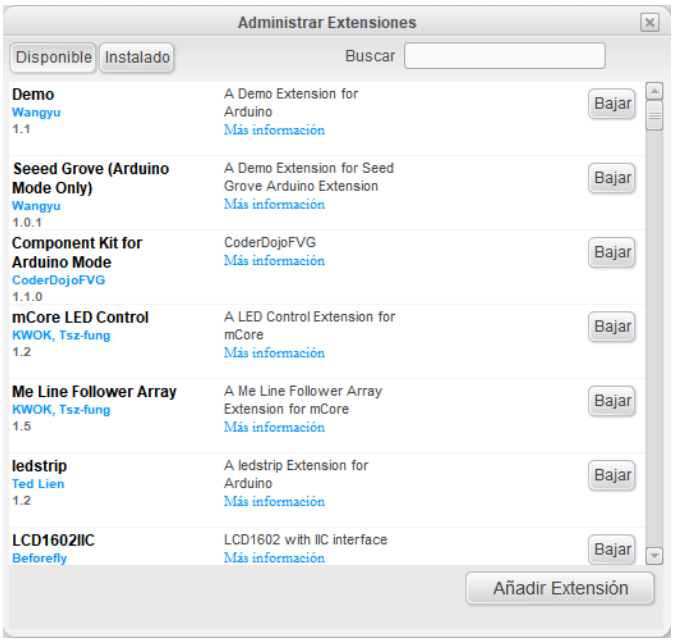



La ventaja de utilizar bloques con instrucciones predeterminada es que simplifican la compresión de los distintos programas y además nos permiten trabajar el pensamiento computacional desde una edad muy temprana ya que no necesitamos escribir código sino ensamblar los distintos bloques que disponemos para conformar nuestro programa, por lo tanto, nos centramos más en cómo debe de realizar nuestro robot la tarea que en como programar la tarea.

La manera de trabajar será por paletas en las cuales tendremos distintos bloques ya creados que podremos combinar hasta obtener el algoritmo deseado.




Al igual que en el Arduino IDE teníamos librerías mblock nos permite descargar extensiones de terceros con lo cual podemos ampliar la gama de sensores y actuadores a utilizar en nuestro entorno.

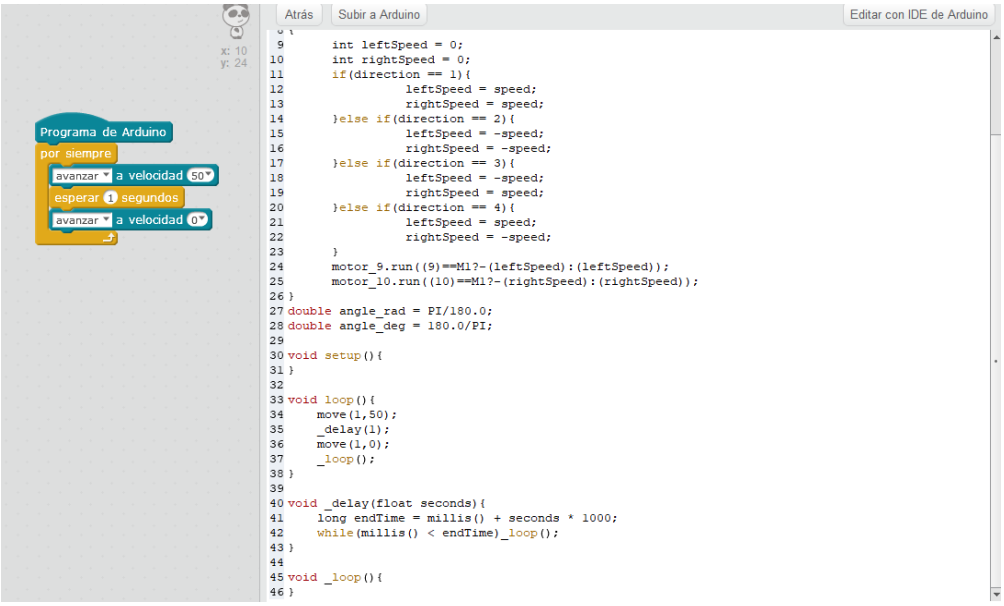



Otra gran ventaja que nos ofrece este entrono es la programación de nuestros robots para que funciones de manera autónoma, es decir podemos programar nuestro robot y no es necesario que esté conectado a nuestro PC.

Para poder programar nuestro robot mBlock se sirve del entorno de Arduino IDE y los bloques que utilizamos se transforman directamente en código escrito que puede ser compilado y programado en nuestras unidades de control.

Por lo tanto, la utilización de cualquier plataforma hardware de Arduino puede ser programada por este entorno de programación.

El siguiente ejemplo nos muestra un programa que mueve un robot 1 segundo hacia adelante creado por bloque y como el entorno lo traduce para su compilación en Arduino.




Como podemos ver en la imagen a la izquierda tenemos el programa realizado mediante bloques y la derecha su traducción para Arduino IDE.

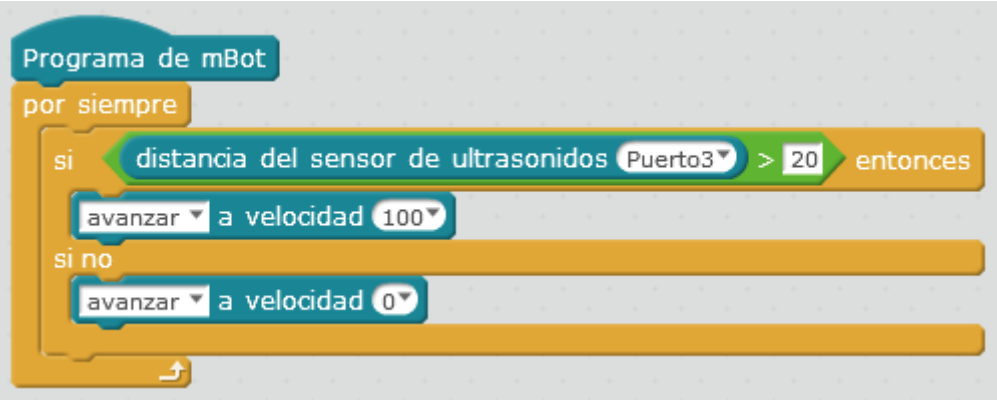
El en siguiente tabla vamos a resumir las principales características de mBlock.

| | |
|---|---|
| **Plataformas permitidas** | Linux, Windows y Mac OS. |
| **S.O Lliurex** | Permite la instalación y la programación en Lliurex. |
| **Precio** | No tiene coste. |
| **Programación** | Código basado en bloques y permite el trabajo autónomo del robot. |
| **Curva de aprendizaje** | Baja. |
| **Librerías** | Trabaja por extensiones (**No disponibles en lliurex**).<br>Sin coste. |

En el siguiente enlace tenéis la página oficial de para la descarga del mBlock.

http://www.mblock.cc/mblock-software/

En el siguiente ejemplo implementaremos mediante bloques en mbot un programa que hará que el robot se para cuando el sensor de ultrasonidos detecte un obstáculo a 20cm.



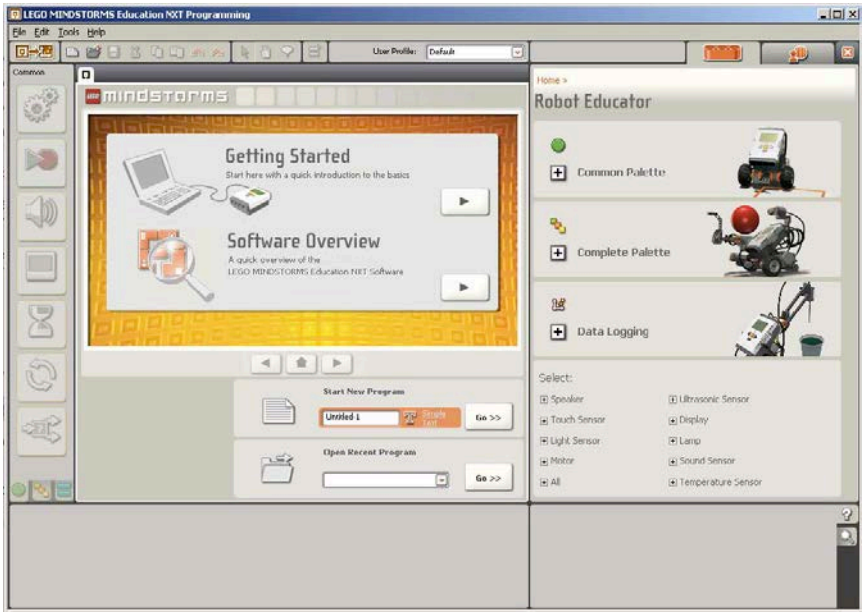

Como podemos apreciar la programación mediante la utilización de bloques es muy sencilla e intuitiva.

## 3. NXT PROGRAMMING.

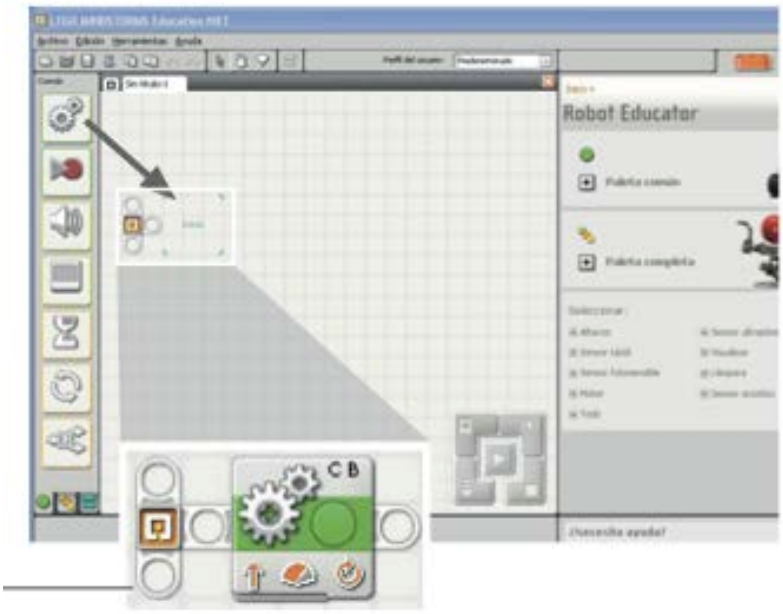
El entorno de programación NXT Programming, de ahora en adelante NXT, es entorno diseñado para la programación de los robots de lego NXT y está basado en sistema de programación visual de Nacional Instruments, LabView.

En este entorno programamos como en caso del entorno mBlock utilizaremos bloques pre programados los cuales uniremos para crear nuestro programa, lenguaje de programación NXT-G.

Ya que este entorno está diseñado específicamente para la utilización del robot educativo NXT los bloques son más robustos que en caso anterior y nos permite configurar los sensores de manera muy sencilla pudiendo controlar muchos parámetros de nuestro sensores y actuadores de manera muy sencilla.




Al contrario que en el entorno de mBlock en cual vamos montando una especie de puzles en la programación de NXT tendremos que unir los bloques de izquierda a derecha y se ejecuta de manera secuencial por lo que facilita la introducción de la programación secuencial.

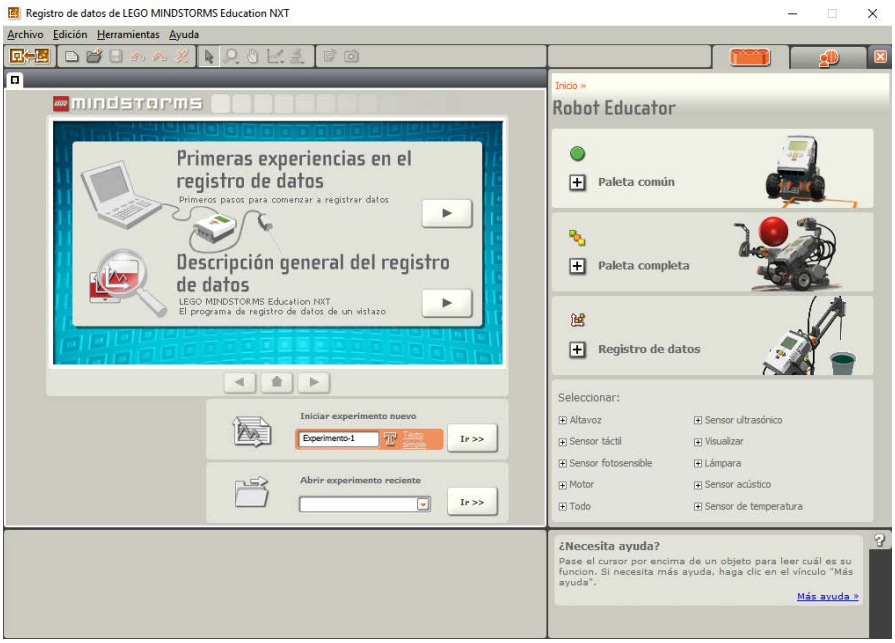



Como podemos ver en la imagen anterior el programa nos solo nos deja poner bloque sobre el tabique (indicado con la fecha) y luego seguir uniendo bloque a la derecha del anterior.

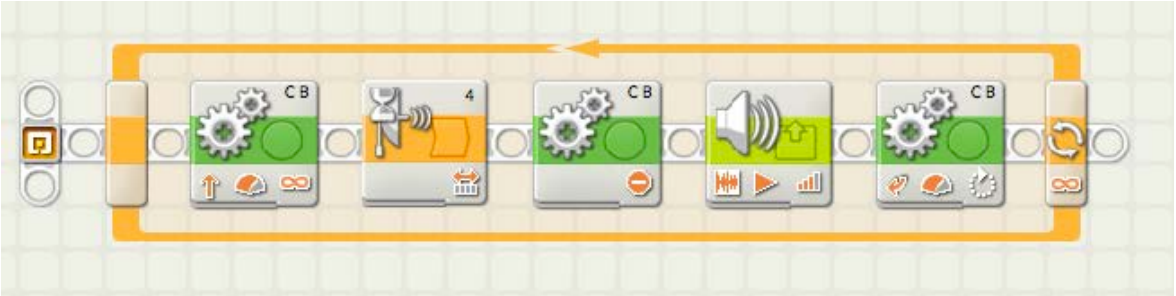
El entorno de NXT también nos proporciona el Robot Educador que es un compendio de montajes y programas que nos permite aprender el funcionamiento y la programación de los distintos sensores y actuadores, la única pega es que la versión con el robot educador nos es gratuita.

El entorno de programación de NXT también nos ofrece una herramienta muy útil que el NXT Data logging.

Esta herramienta nos permite representar los distintos sensores en tiempo real como el almacenamiento de los valores con lo que podremos realizar proyectos no solo orientados a la utilización de un robot sino más trasversales.




El siguiente ejemplo nos muestra un programa mediante el cual el robot avanzará hasta encontrar un obstáculo a cierta distancia, parará, emitirá un sonido y girar 180º para proseguir su camino.



El en siguiente tabla vamos a resumir las principales características de NXT.

| | |
|---|---|
| **Plataformas permitidas** | Windows. |
| **S.O Lliurex** | No permite la instalación y la programación en Lliurex. |
| **Precio** | No tiene coste su versión estudiante. |
| **Programación** | Código basado en bloques y permite el trabajo autónomo del robot. |
| **Curva de aprendizaje** | Baja. |
| **Librerías** | No posee librerías externas y Si bloques. |

# 4. LEGO MINDSTORMS EDUCATION.

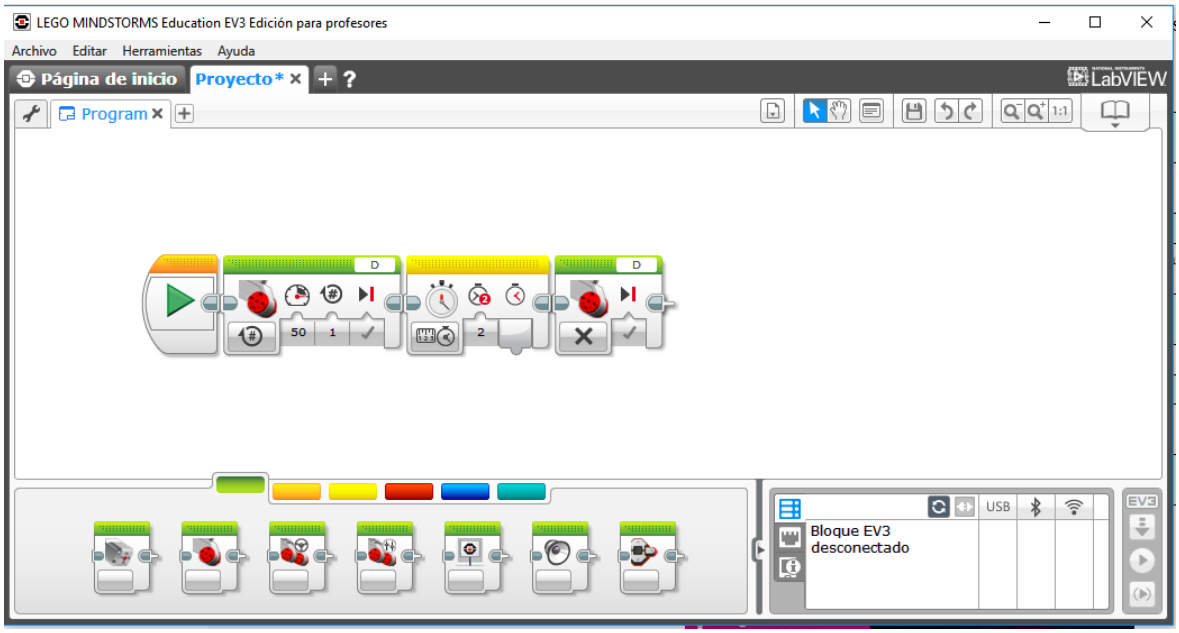
El entorno de programación Lego Mindstorms Education, de ahora en adelante EV3, es entorno diseñado para la programación de los robots de lego EV3 y está basado en sistema de programación visual de Nacional Instruments, LabView.

Este entorno de programación es la evolución del entorno de programación NXT programming y está destinado para la programación del robot EV3 de lego que a su vez es la evolución del robot NXT de Lego.

En este entorno no podremos programar en **ningún caso** la versión del robot NXT de lego, ya que como pasaba en el entorno anterior la programación se realiza por bloques que están diseñados para los sensores y actuadores que posee el robot EV3 específicamente.

Este entorno, aunque sigue la misma filosofía que el entorno de programación de NXT no es tan recomendable para su uso en la etapa inicial de primaria, como puede ser el caso del robot NXT, ya que el gran potencial que nos ofrece este entorno hace que sea más complicado de utilizar.

La siguiente imagen nos muestra el entorno de programación Lego Mindstorms Education.



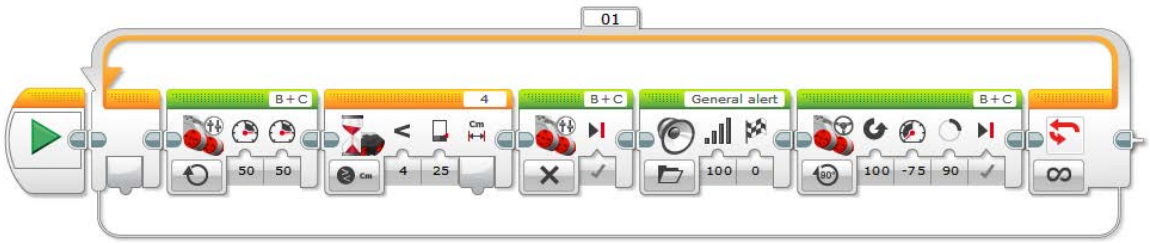

Como ocurría con su predecesor este entorno no solo está pesado para la programación del EV3 como robot, sino que también no ofrece la creación de proyector como experimentos, mediante el cual podemos utilizar los distintos sensores y actuadores además de monitorizar distintos parámetros y realizar estudios de distintas magnitudes físicas y efectos.

Tambien posee la herramienta Robot Educador mediante la cual se nos facilita una gran cantidad de proyectos para realizar con los el EV3 tanto a nivel de montaje como de programación.

El en siguiente tabla vamos a resumir las principales características del entorno EV3.

| | |
|---|---|
| **Plataformas permitidas** | Windows. |
| **S.O Lliurex** | No permite la instalación y la programación en Lliurex. |
| **Precio** | No tiene coste. |
| **Programación** | Código basado en bloques y permite el trabajo autónomo del robot. |
| **Curva de aprendizaje** | Bajo/media. |
| **Librerías** | No posee librerías externas, Si bloques externos. |

El siguiente ejemplo nos muestra un programa mediante el cual el robot avanzará hasta encontrar un obstáculo a cierta distancia, parará, emitirá un sonido y girar 180º para proseguir su camino.




Como se puede apreciar la programación es muy parecida a la del entorno de programación NXT.

# 5. BITBLOQ.

El entorno de programación Bitbloq es un entorno de programación grafico desarrollado por la empresa tecnológica BQ.

Bitbloq puede utilizarse con gran cantidad de plataformas como el robot mBot, sistemas basados en Arduinos, Zoki, Evolution …etc.

Bitbloq es una plataforma web de acceso libre pero el usuario deberá registrarse para poder almacenar los proyectos desarrollados.

También presenta el inconveniente de que la utilización de esta plataforma con robots como el mBot necesitaremos una licencia (normalmente es suministrada con la compra del robot) o comprar una licencia vitalicia que ronda los 4 euros.

Esta plataforma solo es operativa con el navegador Google Chrome además necesitaremos instalar el programa Web2Board para poder conectar con las distintas placas.

La programación se tiene que realizar en dos pasos, primero tendremos un editor en el cual tendremos que definir el hardware de nuestro robot y posteriormente otro menú en podremos crear nuestro programa.

El lenguaje de programación al igual que los casos anteriores se realiza mediante bloques con distintas funciones que iremos apilando para crear nuestro programa.

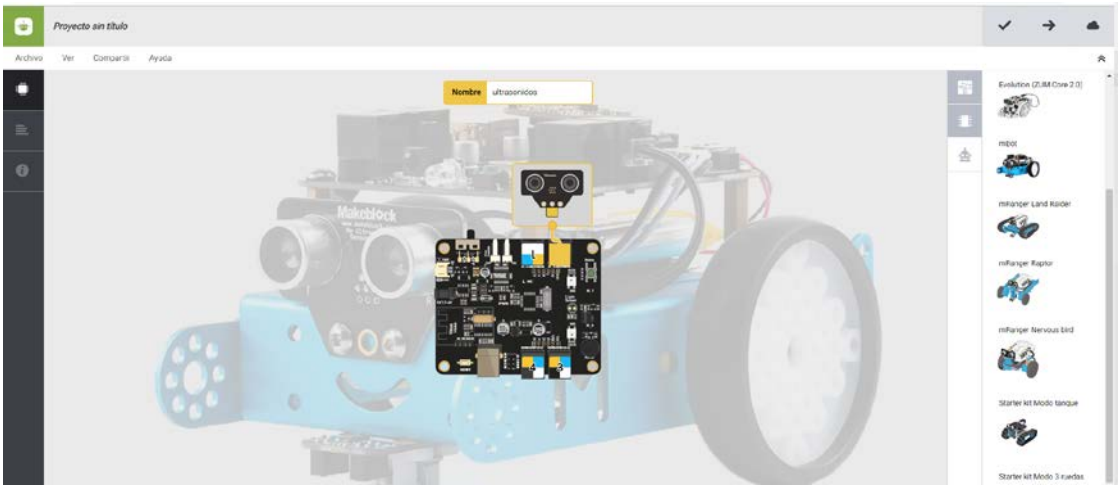
El siguiente enlace nos llevar al programa Bitbloq en nuestro navegador.

https://bitbloq.bq.com/#/

En el siguiente ejemplo vamos a realizar un programa un proyecto sobre la plataforma mBot la cual realizara la siguiente tarea.

Avanzar hasta encontrar un obstáculo a menos de 20cm, emitir un sonido y girar y proseguir avanzando.

Como podemos ver en la siguiente imagen hemos definido una plataforma hardware de mbot con un sensor de ultrasonidos.




En la siguiente figura vemos la pantalla de programación y el código que resuelve el problema planteado.

El en siguiente tabla vamos a resumir las principales características del entorno Bitbloq.

| | |
|---|---|
| **Plataformas permitidas** | Navegador Google Chrome |
| **S.O Lliurex** | Si permite necesario navegador de Google Chrome. |
| **Precio** | Licencia para algunas plataformas . |
| **Programación** | Código basado en bloques. |
| **Curva de aprendizaje** | Bajo/media. |
| **Librerías** | No posee librerías externas. |

# TEMA 5. EJEMPLOS DE PROYECTOS DE ROBÓTICA EDUCATIVA

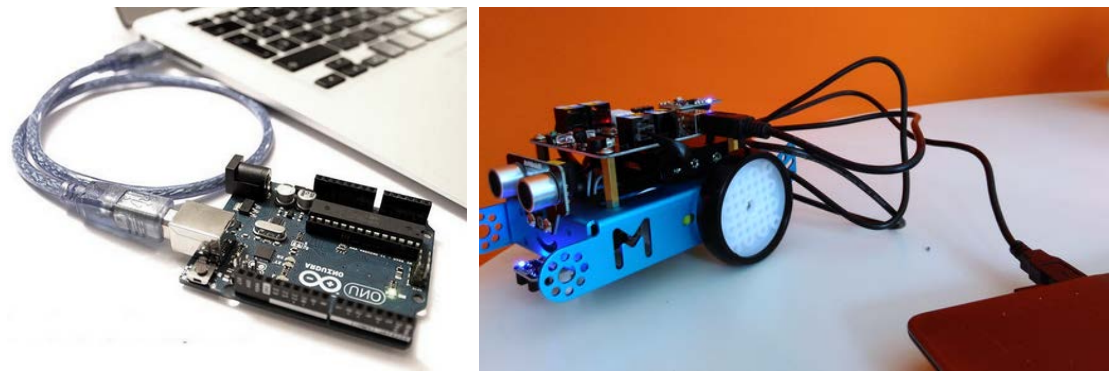
## 1. INTRODUCCIÓN

En este último tema del curso se pretende mostrar una serie de proyectos de ejemplo para introducir a los alumnos y alumnas al uso de robots educativos, partiendo desde ejemplos muy sencillos centrados en familiarizarse con la conexión del robot al ordenador y el manejo de las diferentes partes del robot de forma separada. De este modo, se está dividiendo la tarea de aprender a manejar el robot en subtareas más sencillas con el objetivo de facilitar el aprendizaje.

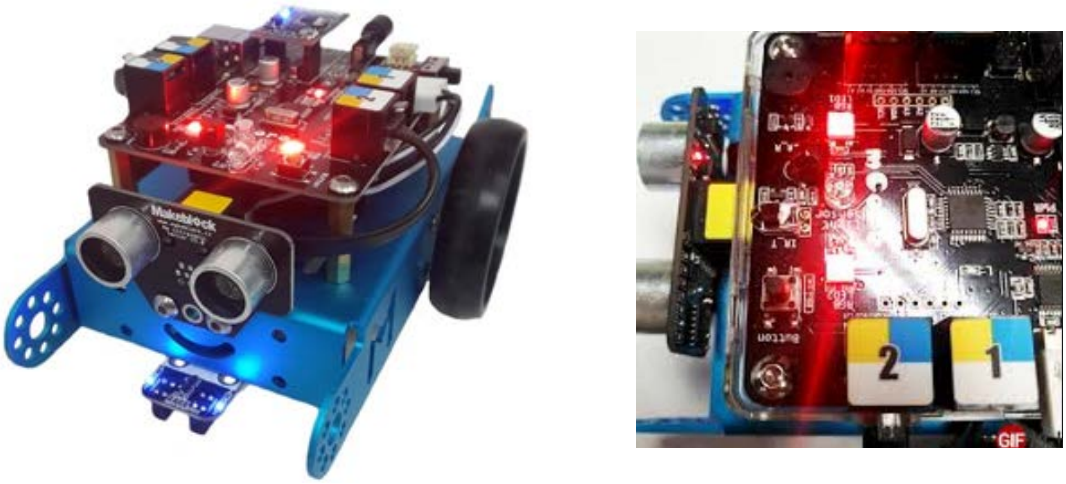
Una vez se ha abordado la programación de las diferentes partes del robot y el alumno o la alumna es capaz de introducir comportamientos sencillos en el robot y controlarlo, será capaz de avanzar en el aprendizaje y tratar de resolver proyectos en los que diferentes partes de las trabajas interactúen entre sí. Es por esto que, la primera parte de introducción debe realizarse planteando unos objetivos realistas y alcanzables, ya que es más interesante que aprendan a manejar dos sensores y dos motores, antes que trabajar un gran abanico de sensores y actuadores, pero no lleguen a comprender su realmente su funcionamiento.

## 2. EJEMPLO 1: PARPADEAR LED

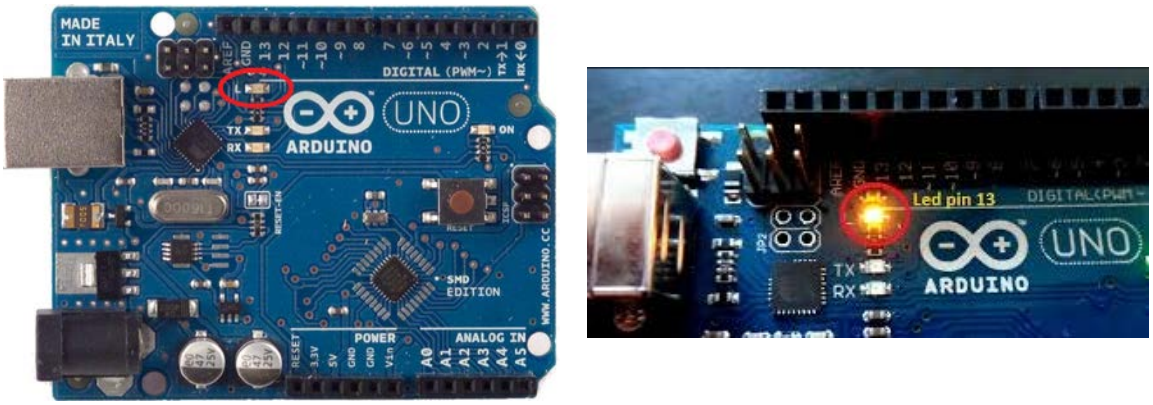
Uno de los programas de introducción más empleados para verificar si existe comunicación entre el ordenador y el robot a emplear es el de hacer que parpadee un LED de los que los robots suele llegar integrado. Esto es debido a la facilidad del programa, ya que el objetivo de este ejemplo no es avanzar en términos de programación, sino partir de la base de que existe una conexión correcta entre el robot y el ordenador. Esta conexión suele realizarse mediante cable USB, aunque también existe la posibilidad de realizarla mediante conexión inalámbrica bluetooth.



*Conexión mediante cable USB entre Arduino Uno (izquierda) y robot mBot (derecha) al PC*




*LEDs encendidos en el robot mBot*



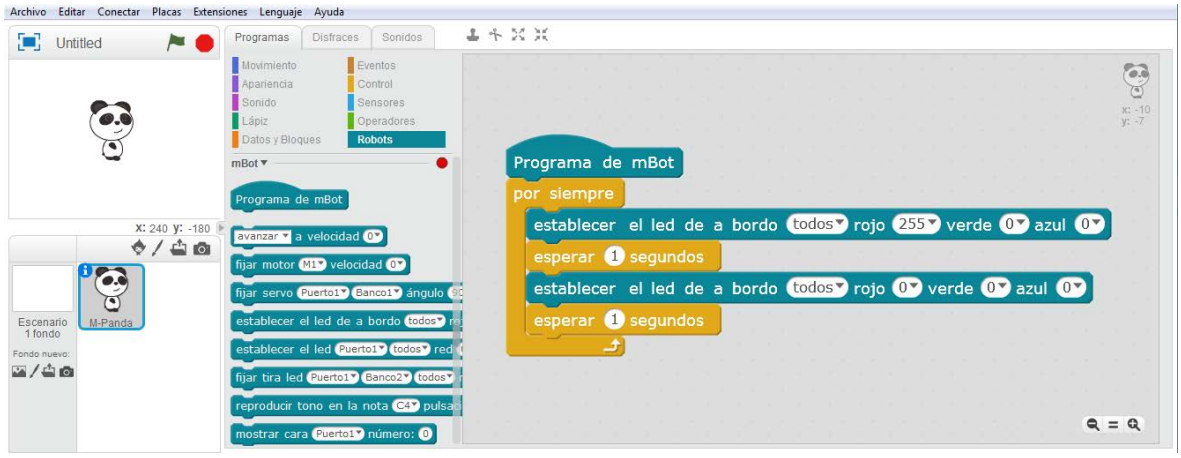

*LED encendido en la placa Arduino Uno*

Para realizar este ejemplo, previamente se tiene que haber realizado una presentación explicando la plataforma de programación que se va a trabajar y los bloques básicos que permiten encender los dos LEDs que integra mBot o activar el pin 13 en el caso de Arduino UNO, ya que el LED que integra en su placa se encuentra internamente conectado al pin 13 de la placa.

Es importante que el enunciado sea claro y lo más sencillo posible, definiendo todos los parámetros a programar. En este caso, el enunciado de este primer ejemplo podría ser el siguiente para el robot mBot:

*Crea un programa en mBlock que haga que los LEDs de a bordo se enciendan mostrando un color rojo durante 1 segundo y, a continuación, se apaguen durante 1 segundo. Repetir esta tarea de forma indefinida.*

La solución de este ejemplo sería la siguiente:




De este modo, con muy pocos bloques es posible programar el comportamiento requerido en el robot y al mismo tiempo es muy sencillo y visual de comprobar.

Si observamos los bloques empleados (de arriba hacia abajo), se puede observar un primer bloque que indica el tipo de robot a programar (en este caso el mBot), a continuación, se ha incluido el bloque “por siempre” que significa que el programa se va a ejecutar un número de veces ilimitado, es decir, el LED rojo va a parpadear cada segundo infinitas veces, hasta que apaguemos el robot.

Dentro del bloque “por siempre” se ha integrado el bloque “establecer el led de abordo” que permite seleccionar qué LED se quiere configurar (en nuestro ejemplo, los dos LEDs de la placa de mBot), se ha seleccionado el LED rojo con un valor de 255 para que tenga la máxima potencia de brillo (valor 0 indica apagado y 255 brillo máximo) y por último se ha mantenido con valor cero los tonos verde y azul del LED para que el color representado sea rojo puro.

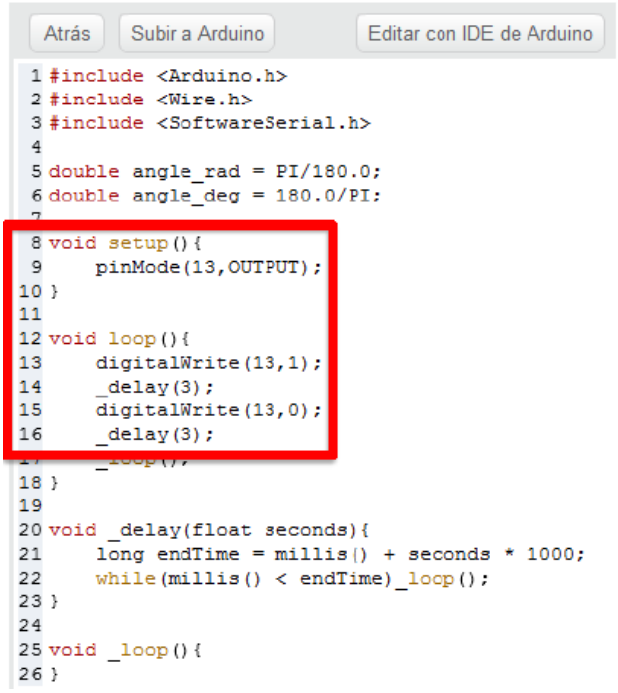
Seguidamente se ha empelado el bloque “esperar X segundos” y se ha configurado con el valor 1 (segundos) para cumplir con el comportamiento solicitado en el enunciado.

Estos dos bloques se han duplicado y únicamente se ha modificado el valor del LED, introduciendo el valor 0 a la componente “rojo”, de modo que los LEDs se mostrarán apagados durante “1 segundo” que es el tiempo de espera configurado en el último bloque.

Con todo esto, se consigue un programa que hace que se enciendan los dos LEDs que incorpora mBot durante un segundo con un tono rojo y que pasado este tiempo se apaguen durante otro segundo, realizando esta tarea infinitas veces.

Para mostrar cómo se realizaría este mismo comportamiento empleando la placa Arduino UNO y la misma plataforma de programación (mBlock), pero empleando los bloques de Arduino en lugar de mBot. El enunciado sería el siguiente:

*Crea un programa en mBlock que haga que el LED de a bordo se encienda durante 3 segundos y, a continuación, se apague durante 3 segundos. Repetir esta tarea de forma indefinida.*


En este caso, también se necesitan muy pocos bloques para realizar la programación, pero es necesario saber que para encender un LED se necesita emplear un pin digital y el número de pin al que se encuentra conectado dicho LED, por lo que, la complejidad aumenta con respecto a la plataforma mBot. La ventaja de trabajar con mBlock en modo Arduino es que aparece una ventana en el entorno de programación que permite observar el código que se genera a partir de los bloques empleados. Esta opción puede resultar interesante para cursos superiores en el que es posible empezar a trabajar con programación por código.

Si observamos los bloques empleados (de arriba hacia abajo), se puede observar que la estructura es muy similar a la empleada para mBot, con un primer bloque que indica el tipo de robot a programar (en este caso Arduino), a continuación, se ha incluido el bloque “por siempre” que significa que el programa se va a ejecutar un número de veces ilimitado, es decir, el LED va a parpadear cada tres segundos infinitas veces, hasta que desconectemos la placa Arduino.

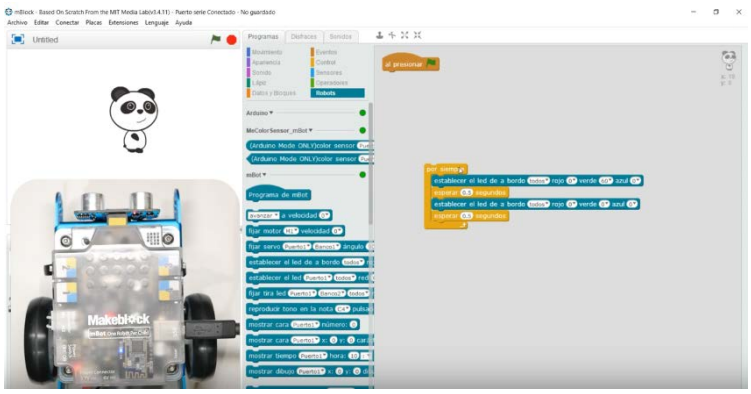
Dentro del bloque “por siempre” se ha integrado el bloque “fijar salida pin digital” que permite seleccionar el pin al que se encuentra conectado el LED integrado en la placa de Arduino y el estado del pin, en este caso “ALTO” para encender el LED. Se ha seleccionado el pin número 13 y en este caso el LED es mono color y no es necesario indicar el tono.

Seguidamente se ha empelado el bloque “esperar X segundos” y se ha configurado con el valor 3 (segundos) para cumplir con el comportamiento solicitado en el enunciado.

Estos dos bloques se han duplicado y únicamente se ha modificado el valor del pin digital, introduciendo el valor “BAJO”, de modo que el LED se mostrarán apagados durante “3 segundos” que es el tiempo de espera configurado en el último bloque.

Con todo esto, se consigue un programa que hace que se encienda el LED que incorpora Arduino UNO durante tres segundos y que pasado este tiempo se apaguen durante otros tres segundos, realizando esta tarea infinitas veces.

Este es un programa muy sencillo de programar, pero que además de verificar la conexión y la correcta instalación del entorno de programación y el estado del robot, le aporta al estudiante una seguridad y una motivación necesario para continuar aprendiendo a programar comportamientos más complejos en el robot. En el siguiente vídeo se puede observar un comportamiento similar al descrito en este ejemplo empleando mBot:



https://youtu.be/oCE9VGJq1cw?t=328

## 3. Ejemplo 2: Hola Mundo

Otro ejemplo muy empleado en el ámbito de la programación y la robótica para verificar el funcionamiento de los robots es el del conocido como “Hola Mundo”. En informática, un programa “Hola Mundo” es el que imprime el texto “Hola mundo!” en un dispositivo de visualización, en la mayoría de los casos una pantalla de ordenador al que se encuentra conectado el robot o la propia pantalla del robot. Este programa suele ser usado como

introducción al estudio de un lenguaje de programación, siendo un primer ejercicio típico, y se considera fundamental desde el punto de vista didáctico.

Es importante que el enunciado sea claro y lo más sencillo posible, definiendo todos los parámetros a programar. En este caso, el enunciado de este primer ejemplo podría ser el siguiente para el robot mBot:

*Crea un programa en mBlock para que el robot mBot envíe la frase "Hola Mundo!" cada dos segundos un total de cinco veces por comunicación serie (a través del cable USB o mediante conexión bluetooth) al ordenador.*

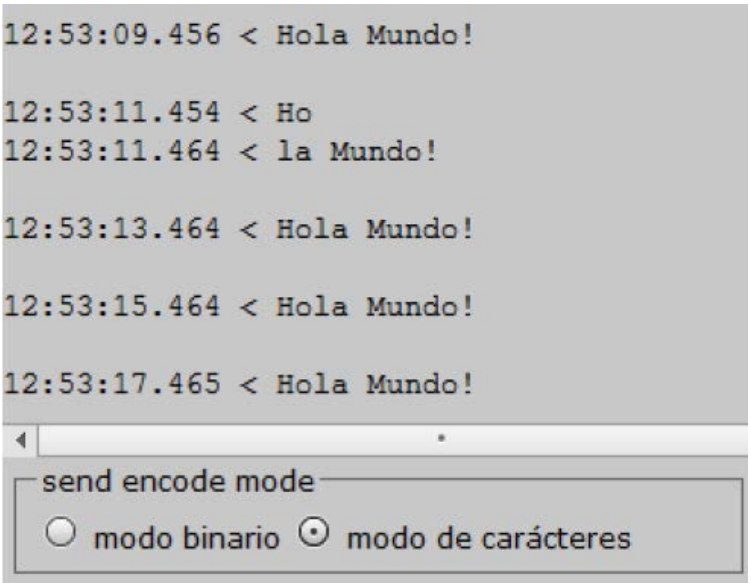

Del mismo modo que en el ejemplo anterior, en primer lugar se selecciona como bloque principal el que indica el tipo de robot o placa se está empleando, para este ejemplo, el robot mBot.

A continuación, como se indica que debe realizarse el envío de datos únicamente 5 veces (y no ilimitadamente como ocurría en el programa de parpadeo del LED), se ha empelado el bloque "repetir X" y se ha configurado con un valor 5.

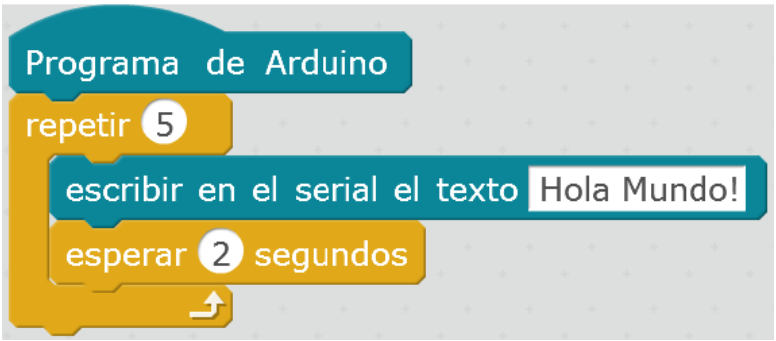
Dentro de este bloque se ha integrado en primer lugar el bloque "escribir en el serial el texto X" y se ha escrito la frase que indica el enunciado (Hola Mundo!) para enviar el mensaje desde el robot al ordenador mediante comunicación serial o USB y ser representado en la plataforma mBlock.

Por último, se ha incluido un bloque de "esperar X segundos", introduciendo el valor 2 como se especifica en el enunciado.

Una vez programado el robot, en una de los cuadros que aparecen en la plataforma de programación mBlock, se podrá visualizar como se recibe la frase “Hola Mundo!” enviada por el robot mBot cada 2 segundos un total de cinco veces.




Para programar este mismo comportamiento en la placa Arduino UNO empleando la plataforma de programación mBlock, bastaría con cambiar el bloque principal que indica el robot o placa que se está utilizando, en este caso “Programa de Arduino” en lugar de mBot:




Este comportamiento que permite enviar información desde el robot al ordenador (y viceversa) es muy útil cuando se desea monitorizar los valores que proporcionan determinados sensores, para verificar que funcionan y que se ha programado el comportamiento deseado correctamente. Por ejemplo, si se desea monitorizar la temperatura de una sala durante 24 horas y se emplea un robot que mide la temperatura cada 15 minutos y la envía al ordenador, finalizadas las 24h de medición, podrían copiarse los datos enviados por el robot y copiarse en una hoja de cálculo para representar las variaciones de temperatura durante el periodo de adquisición de datos.

## 4. Ejemplo 3: Manejo del Robot

El tercer ejemplo a trabajar con los alumnos y las alumnas está relacionado con el manejo y control del robot, en el caso de que éste presente una estructura de robot móvil. Es importante trabajar estos tres ejemplos de programación de forma guiada para que los estudiantes comprendan correctamente el proceso de conexión con el robot, programación de comportamientos y respuesta del robot.

Para este ejemplo se continúa tomando como ejemplo los robots mBot y la placa Arduino Uno de forma que se pueda observar la comparativa en términos de complejidad de programación de una plataforma comercial (mBot) con una puramente libre (Arduino) empleando la misma plataforma de programación por bloques (mBlock).

En primer lugar, hay que tener en cuenta cómo están conectados los motores a la placa del microprocesador (mCore en el caso de mBot). En las instrucciones de montaje de mBot, se especifica que M1 es el motor izquierdo y M2 el motor derecho. Si esta conexión se alterna, los bloques que se empleen durante la programación no realizarán el comportamiento deseado.

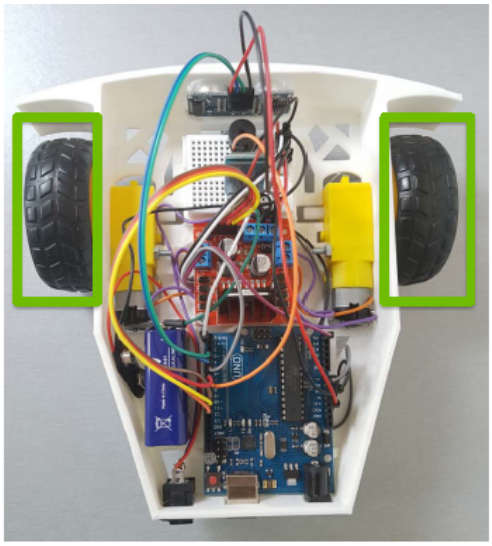

En este tipo de plataformas en las que existen dos ruedas motrices y una rueda loca o libre que se emplea como tribote para seguir el movimiento de las motrices, por lo que para avanzar hacia adelante o hacia detrás es necesario que ambos motores giren en el mismo sentido (horario o antihorario dependiendo el sentido deseado). Para realizar un giro, es posible mover un único motor, por ejemplo, si se desea girar a la izquierda el robot, es necesario apagar el motor izquierdo y encender el motor derecho. Si se desea que el robot gire sobre sí mismo, hacia la izquierda se debería retroceder con el motor izquierdo y al mismo tiempo avanzar con el derecho.

Si se va a trabajar a nivel de activación de motores de forma individual, como puede realizarse con mBot y como se suele hacer con Arduino, lo más sencillo es realizar una tabla de verdad en la que se recojan los diferentes casos y se indique qué motor debe activarse en cada uno de ellos. De este modo, los alumnos pueden comprender con mayor facilidad cómo se programan los posibles movimientos. A continuación, se muestra un ejemplo para un robot casero construido empleando un Arduino UNO, conectando el motor derecho a los pines 9 y 6 y el motor izquierdo a los pines 11 y 10. Para que exista movimiento en un motor debe existir una diferencia de tensión entre sus dos pines, es decir, para mover el motor derecho hacia delante es necesario configurar el pin 9 a nivel alto (5V = "1") y el pin 6 a nivel bajo (0V = "0").

**RUEDA IZQUIERDA**
Controlada por
PIN 11 y PIN 10

**AVANZA** si
PIN 11=ALTO
y
PIN 10=BAJO

**RETROCEDE** si
PIN 11=BAJO
y
PIN 10=ALTO



**RUEDA DERECHA**
Controlada por
PIN 9 y PIN 6

**AVANZA** si
PIN 9=ALTO
y
PIN 6=BAJO

**RETROCEDE** si
PIN 9=BAJO
y
PIN 6=ALTO

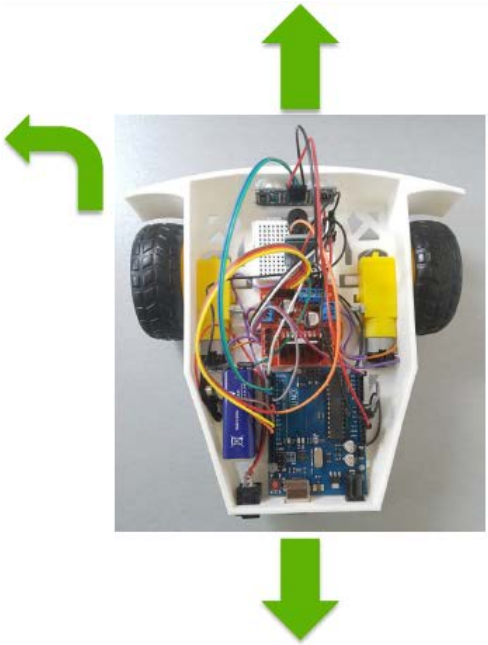
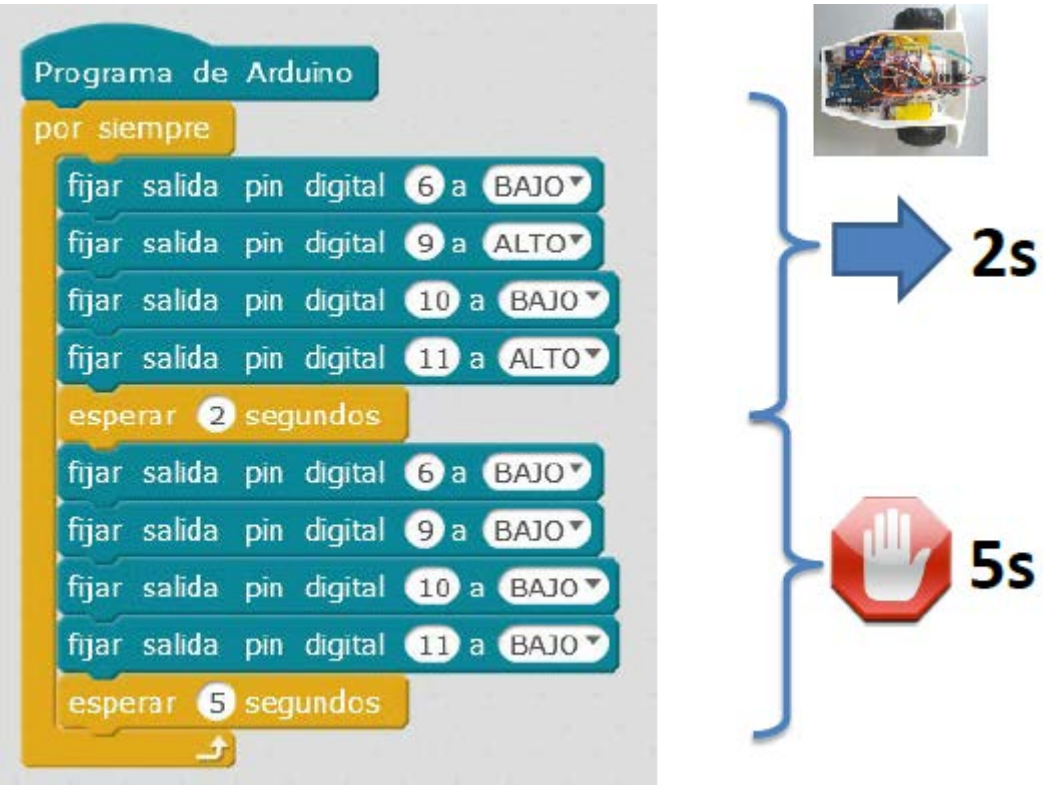


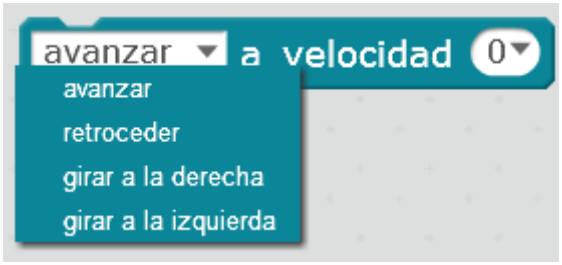
**TABLA DE VERDAD:**
**MOVIMIENTOS DEL ROBOT**

| | PIN11 | PIN10 | PIN9 | PIN6 |
|---|---|---|---|---|
| AVANZAR | 1 | 0 | 1 | 0 |
| RETROCEDER | 0 | 1 | 0 | 1 |
| GIRO DERECHA | 1 | 0 | 0 | 0 |
| GIRO IZQUIERDA | 0 | 0 | 1 | 0 |

Un ejemplo de programación de Arduino UNO empleando mBlock en el que el robot avanza 2 segundos y se mantiene parado durante 5 segundos de forma ilimitada puede verse en el siguiente código:




En el caso de mBot, la gestión de los motores y en definitiva de los diferentes movimientos del robot puede realizarse de forma mucho más sencilla mediante un único bloque que permite indicar si se desea que el robot avance, retroceda, gire a la derecha o a la izquierda y con qué potencia (donde 0 es parar y 255 es el máximo de potencia de los motores):




El mismo programa programado para Arduino Uno en el que el robot avanza 2 segundos y se mantiene parado 5 segundos de forma ilimitada se programaría de la siguiente forma para el robot mBot:


Asimismo, para niveles educativos superiores también podría programarse a nivel de motor consiguiendo el mismo comportamiento, pero ahondando un nivel más en la complejidad del programa (sin llegar a ser tan complejo como en Arduino Uno):


## 5. Ejercicio con un Sensor: Sigue Luz

Una vez expuestos los tres ejemplos guiados que permiten controlar familiarizarse con la plataforma de programación, los bloques de comportamiento, interconexión del robot y control y comunicación del mismo, es posible empezar a plantear problemas a resolver relacionados con la adquisición de información mediante sensores, una posible toma de decisión por el procesador y una posterior reacción mediante los actuadores. Los más sencillo es empezar con un sensor lineal que proporcione un valor de variable mayor conforme aumente la magnitud a medir. Por ejemplo, un sensor sencillo de controlar es el de luz (LDR) de forma que a mayor luz medida la variable a analizar proporciona un valor superior (entre 0 y 1023 habitualmente).

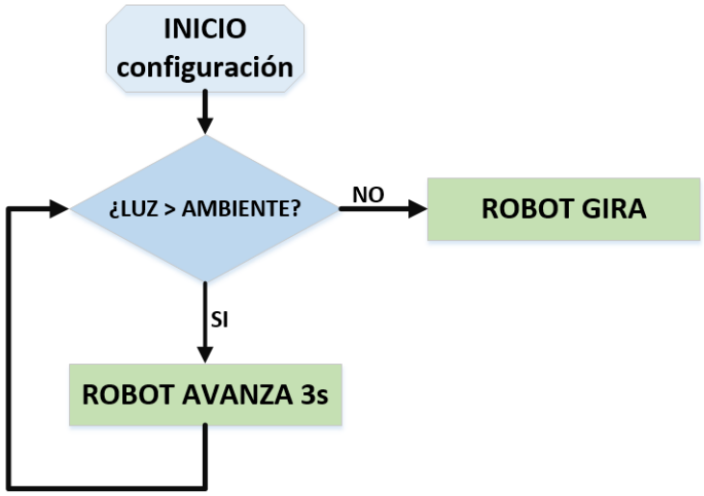
Para favorecer la estrategia de motivación es necesario que se planteen retos a resolver mediante un enunciado que presente un problema real. En este sentido, en lugar de plantear la actividad como una actividad con el robot en la que se usa el sensor de luz, se podría emplear el siguiente enunciado:

*Se desea programar un sistema de alarma basado en un robot guardián que se encuentre vigilando continuamente todo su perímetro girando sobre sí mismo en sentido horario. Se mantendrá realizando este movimiento hasta que detecte una luz intensa, en ese momento se dirija hacia ella durante 3s. Pasados los 3s, el robot continuará con la vigilancia girando sobre sí mismo.*

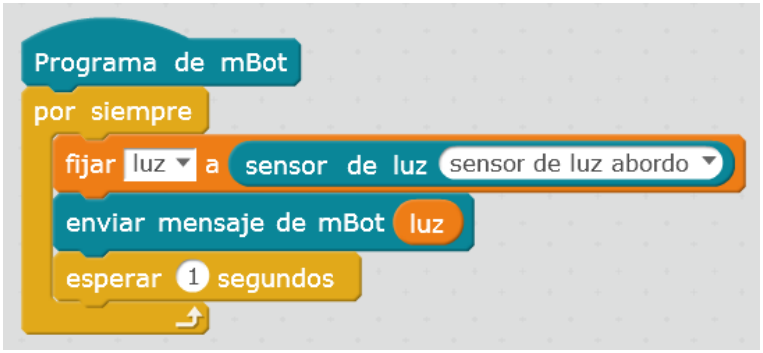
*Para ello, tenemos que tener instalado en el robot, el sensor de luz basado en una fotorresistencia LDR que proporcione el valor de luminosidad que mide. Cuando procesador lee este valor decide si el robot tiene que avanzar o permanecer girando.*


Dependiendo del nivel educativo del alumnado al que se esté dirigiendo la actividad, se podrán facilitar ayudas, como indicar qué bloques conforman la estructura del programa o qué bloque debe emplearse para manejar el sensor de luminosidad. No obstante, cabe recordar que una de las habilidades que se desean trabajar es la de la investigación, por lo que podrían utilizar los recursos que disponen en el PC (Internet, Tutoriales, Videotutoriales…) para resolver el problema planteado.

Una herramienta muy útil para resolver este tipo de actividades es la realización de un diagrama de flujo que valore las posibles variantes del comportamiento del robot:

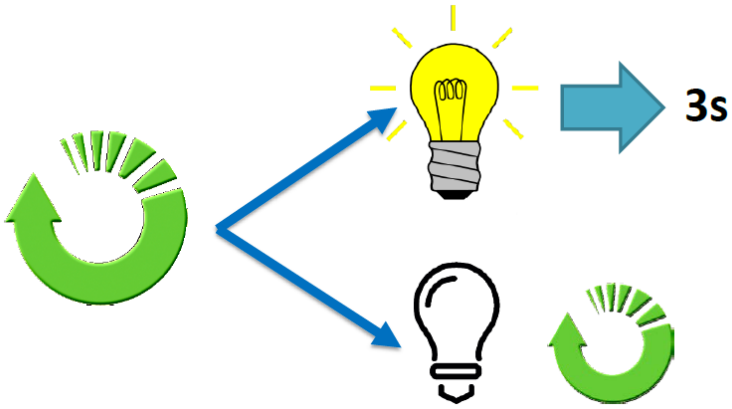



Una primera aproximación podría basarse en conocer el valor de referencia de la luminosidad que hay en la sala y, a partir de ese valor, determinar un valor umbral (suele estar en torno a 600) a partir del cual se considera que se ha excedido el nivel de luz por el cual salta la alarma y el robot debe de parar de girar y empezar a avanzar. Para ello, en el caso de mBot se podría utilizar el siguiente programa que envía el valor de la luz (almacena el valor de sensor en la variable creada con el nombre "luz") medida cada segundo a través del cable USB al ordenador:




En el caso de utilizar un robot con una placa Arduino Uno el programa sería similar, pero habría que indicar en qué pin de la placa de Arduino se ha conectado el sensor LDR. En este caso se ha conectado en el pin A1:

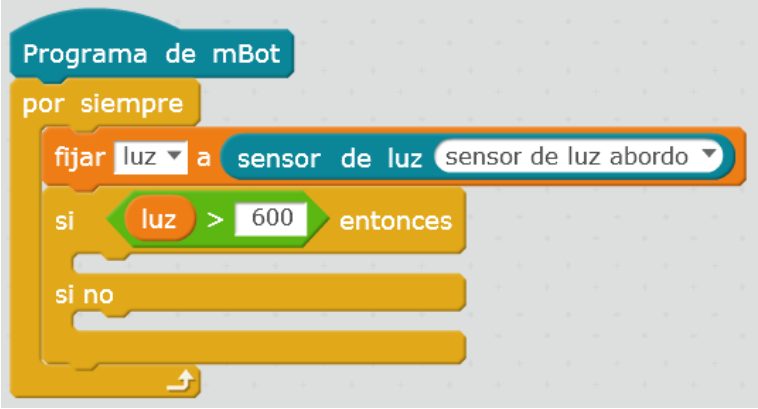

De este modo, con el valor de luz que el robot ha enviado y mostrado en el PC, ya es posible fijar un valor de referencia o umbral en el programa para decidir si la luz medida por el robot es superior a ese umbral y en tal caso el robot avanzaría durante 3 segundos:



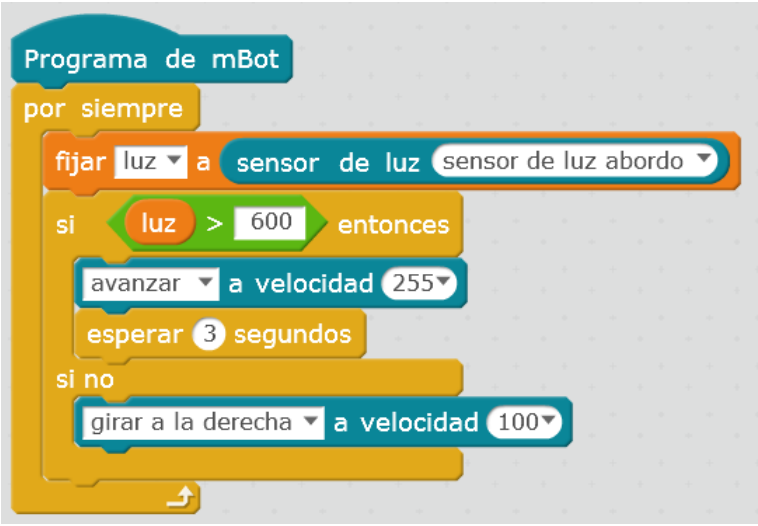

Para que el robot sea capaz de tomar esta decisión, es necesario emplear el bloque SI/ENTONCES que se encuentra dentro del submenú de CONTROL (bloques amarillos).


Entre “si” y “entonces” se debe introducir un operador comparativo, que en este caso será el signo mayor que “>” (podría ser también “<” cambiando la lógica) que se encuentra dentro del submenú OPERADORES (bloques verdes) y habría que comprobar si la variable “luz” que se está midiendo con el sensor es superior al valor de referencia o umbral que se ha medido anteriormente:

Suponiendo que se ha medido un valor umbral de 600, se indica este valor dentro del operador comparativo y se compara con la variable “luz” que indica el valor medido con el sensor en cada momento:




Una vez programada la estructura del programa, se debe incluir el comportamiento de los actuadores, en este caso de los dos motores. Si se cumple que “luz” es superior al valor de la luz ambiente de la sala (luz SÍ es mayor que 600) entonces se activa la alarma, es decir, el robot avanza durante 3 segundos. Si no se cumple continúa girando:




Al programarlo con el robot mBot es posible emplear el bloque que permite indicar el tipo de movimiento (avanzar o girar), en el caso de Arduino sería necesario indicar en cada caso la configuración de los pines conectados a cada uno de los dos motores.

Es interesante en ocasiones dejar algunas variables sin especificar en el enunciado para que el estudiante pueda por sí mismo aprender a través de la experiencia. A partir del enunciado planteado deberían llegar a una estructura de solución muy similar a la planteada, pero deberán justificar los valores introducidos en la potencia de los motores. En la solución presentada, se configuran con máxima potencia (255) al avanzar, dado que se trata de una alarma y el robot debe de responder lo más rápido posible, pero en el caso de la potencia que

indica la velocidad con la que gira el robot, se ha configurado con potencia 100 (potencia media). Este parámetro se ha configurado de este modo para que si el robot se ilumina con una linterna, avance en esa dirección, ya que si gira demasiado rápido se pasará la posición a la que ha detectado la luz y no avanzará en la dirección adecuada.

## 6. Ejemplos de Proyectos

Los ejemplos y la actividad analizadas anteriormente pueden resultar interesantes desde el punto de vista de la asignatura de Tecnología o de Informática, pero en este curso se pretende ir un paso más allá, empleando la robótica de forma transversal aplicándola en otras disciplinas. De este modo, en este último punto se recogen diferentes propuestas de proyectos interdisciplinares que pueden llevarse a cabo empleando el robot como herramienta de trabajo y elemento motivador.

### Robot Barredora Autónoma

Este proyecto está enfocado mayoritariamente a la asignatura de Tecnología o/y de Informática, pero pueden introducirse variantes que permitan relacionarlo con las Matemáticas, definiendo comportamientos que impliquen avanzar distancias concretas (midiendo diámetro de la rueda y transformando movimiento circular en desplazamiento lineal) o girar un determinado ángulo. Cabe destacar que el éxito de un robot barredor autónomo frente a otro suele deberse más a la programación de un algoritmo que permita una limpieza más eficiente que a elementos de la electrónica que integra.

Un posible enunciado para este tipo de reto podría ser el siguiente:

*Un objeto tecnológico que cada vez es más común ver en las casas o en algunos establecimientos es la barredora doméstica automática. Este robot inteligente realiza un aspirado y barrido del suelo de una vivienda o comercio de forma autónoma sin necesidad de supervisarlo debido a la cantidad de sensores y actuadores que dispone. En esta actividad se van a emular algunos de los comportamientos de este dispositivo en el robot. A modo de ejemplo, en la Figura se representa una de las funcionalidades que posee una barredora doméstica automática:*


*Empleando el sensor de ultrasonidos y caracterizando los motores que integra el robot a emplear consigue que:*

- *El robot retroceda 10 centímetros cuando detecte un objeto en la parte frontal a menos de 15 cm.*
- *Una vez retroceda los 10 centímetros, el robot gire 45º en sentido horario.*
- *Una vez realizado el giro, continúe avanzando si no detecta ningún objeto en su parte frontal a menos de 10 cm.*
- *El robot debe realizar este comportamiento de forma ilimitada.*

*Dibuja el diagrama de flujo y programa los comportamientos especificados por el fabricante.*

## ROBOT QUIZ

Este proyecto es transversal a cualquier materia, ya que consiste en programar varios robots y dividir la clase en varios grupos, asignando cada uno de ellos a un robot. Dentro de los robots se programan varias preguntas de forma que las represente a través del ordenador mediante comunicación serie (USB) o de una pantalla si dispone de la misma.

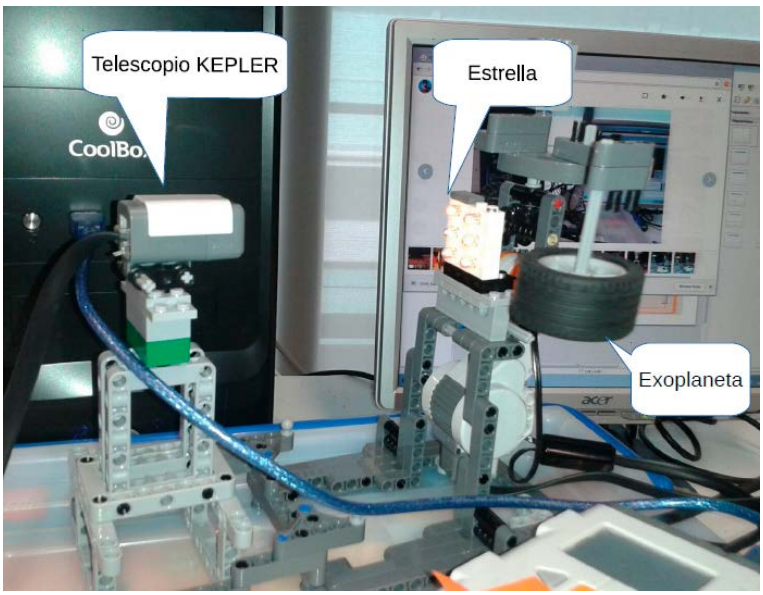
Cada pregunta se mostrará durante 60 segundos y los miembros de cada equipo asignados a cada robot tendrán que presionar alguno de sus botones para contestarla de modo que cada botón corresponde a una respuesta. Es decir, la pregunta tipo test podría tener tres respuestas y se asignaría un pulsador a cada respuesta de forma que el grupo pulsaría uno de los tres pulsadores y el robot indicaría si la respuesta es correcta o no mediante el encendido de un LED o de un LED  de determinado color.

El grupo que más aciertos acumule ganaría el juego. Por tanto, los alumnos serían partícipes de una actividad didáctica que ellos mismos han programado.

Otra variante posible podría estar basada en que el robot avanzara si se acierta la pregunta o retroceda si se falla, de forma que el primer robot que pase la meta gana.

## ROBOT ECLIPSE

Este proyecto está enfocado a trabar el fenómeno de los eclipses mediante la medida de luminosidad durante un cierto tiempo durante el que un motor está realizando el giro de un elemento que se interpondrá entre el foco de luz y el sensor de luminosidad. De este modo, es posible realizar cálculos del tiempo que tarda en dar la vuelta el elemento que representa un planeta o satélite, en torno al elemento luminoso que representa una estrella.




## MONITORIZACIÓN Y CONTROL DE UN MINI-INVERNADERO

En este proyecto se lleva a cabo la realización a escala del concepto de un entorno de cultivo automatizado mediante múltiples sistemas de monitorización y ejecución que aseguran un entorno ideal y controlado para permitir el desarrollo óptimo de las plantas, pudiendo simular su hábitat natural estableciendo las condiciones de humedad, riego y temperatura ideales para la especie. Esto lo convierte en una potente herramienta para el cultivo de plantas con necesidades especiales de algunos de estos factores.

Este sistema permite establecer las condiciones del interior del invernadero a las necesidades del cultivo pudiendo simular condiciones atmosféricas de cualquier lugar del mundo mediante el control de las variables de:

- Temperatura.
- Humedad
- Luminosidad.

Para controlar las condiciones del invernadero es necesario el empleo de sensores que permitan monitorizar las variables enumeradas anteriormente y para ello se emplean diferentes sensores. Estos sensores proporcionan la información al microprocesador Arduino para que dedica si se debe activar alguno de los actuadores que regulen las condiciones climatológicas del invernadero de acuerdo con el código programado.

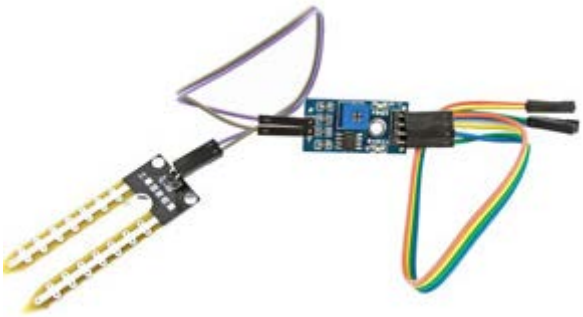
**Controlador, Sensores y Actuadores**

Este tipo de sistema puede ser controlado por una placa microcontroladora Arduino UNO, ya que es una de las soluciones basadas en Arduino más económicas y permite emplear los sensores y actuadores necesarios para implementar este proyecto.

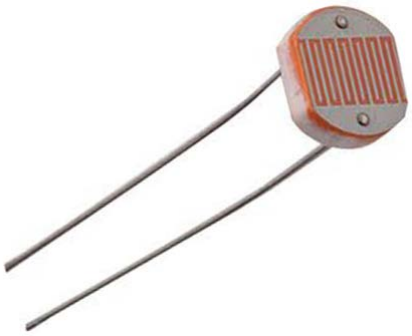

*Placa controladora Arduino UNO*

La placa Arduino UNO recibe la información del sensor de temperatura, el valor de luminosidad de la LDR (del inglés, Light Dependent Resistor) y de sensor de humedad del suelo.

*Sensor de Temperatura LM35*



*Sensor de Humedad de la Tierra*



*Sensor de Luminosidad*

Con estos datos de monitorización, la placa Arduino UNO es capaz de ajustar las condiciones del invernadero mediante diferentes actuadores como una lámpara de calor, un ventilador de extracción de aire o sistema de riego. Estos actuadores pueden estar representados por bombillas LED que emulen el funcionamiento de una lámpara de calor, o por una lámpara de calor real que pueda ser activada o desactivada mediante un dispositivo relé. El ventilador (o motor con aspas) puede estar controlados directamente por Arduino y el sistema de riego puede simularse y realizarlo de forma manual o implementarlo a través de una minibomba de agua que active o desactive el paso de agua a un sistema de goteo.

*LED*


*Ventilador 5V*

*Minibomba de agua*

De este modo, cuando la LDR detecta oscuridad, la placa controladora enciende la luz para aumentar la luminosidad. Si el sensor de temperatura detecta un valor diferente al deseado, se activa el de ventilación (o calefacción si lo hubiera) En cuanto al sensor de humedad de tierra, se podría indicar que es necesario regar mediante un indicador luminoso o sonoro. Si se implementara sistema de riego autónomo se encendería la minibomba de agua hasta que se alcanzara la humedad deseada en la tierra.

Todos estos valores de monitorización y estado de los actuadores podrían representarse en tiempo real a través del monitor serie de Arduino en la pantalla de un PC mediante comunicación serie a través de un cable USB.

Este proyecto podría ser ajustado en términos de dificultad dependiendo del nivel educativo al que vaya dirigido. Una forma sencilla de fijar la complejidad es la introducción o eliminación de elementos en el sistema. Una versión básica de este invernadero podría consistir en la monitorización únicamente del sensor de temperatura y del control del ventilador. Si se deseara aumentar la complejidad, se podrían añadir más bloques como un sensor de lluvia que permitiera conocer si se ha activado correctamente el riego, un motor

que permitiera abrir el techo del invernadero o un sensor que midiera el nivel de humedad del aire.

Siempre es interesante hacer una previsión de los costes del proyecto, ya que la mayoría de componentes compatibles con Arduino son de bajo coste, pero conviene realizar un estudio previo a la definición final del proyecto.

En el caso de este último proyecto, el coste aproximado sería el siguiente:

| Componente | Coste (€) |
|---|---|
| Kit Arduino UNO (LDR, Potenciómetro, Pulsador, Ultrasonidos) | 16,5 |
| Sensor Temperatura LM35 | 1,66 |
| Sensor Humedad de la tierra | 0,98 |
| Mini Bomba Sumergible de Agua DC | 2,99 |
| Sensor Lluvia Detector de Gotas Agua | 1,62 |
| Sensor Nivel de Agua | 1,36 |
| Sensor de Humedad y Temperatura DHT11 | 2,11 |
| Conjunto de cables | 5,26 |
| | **32,48** |