

ROBÓTICA Y TECNOLOGÍA

2026



Julia A. Pérez - Eva F. Giménez

 SEGUNDO AÑO

MÓDULO 1

NOCIONES DE PROGRAMACIÓN

Los Algoritmos permiten describir claramente una serie de instrucciones que debe realizar el computador

para lograr un resultado previsible. Vale la pena recordar que un procedimiento de computador consiste de una serie de instrucciones muy precisas y escritas en un lenguaje de programación que el computador entienda como lo es Scratch.

En resumen, un Algoritmo es una serie ordenada de instrucciones, pasos o procesos que llevan a la solución de un determinado problema. Los hay tan sencillos y cotidianos como seguir la receta del médico, abrir una puerta, lavarse las manos, etc; hasta los que conducen a la solución de problemas muy complejos.

mBlock



mBlock es un entorno de programación gráfico basado en Scratch 3.0, diseñado para facilitar el aprendizaje de la programación y la robótica en entornos educativos. Este entorno permite programar placas Arduino, robots Makeblock, micro:bit y otras plataformas mediante bloques visuales

El entorno de programación de mBlock incluye bloques agrupados por categorías como Movimiento, Apariencia, Sonido, Lápiz y Control, facilitando la creación de proyectos interactivos, animaciones, juegos y programas para robots.

Los robots Makeblock se programan con el software propio de la compañía, llamado mBlock.

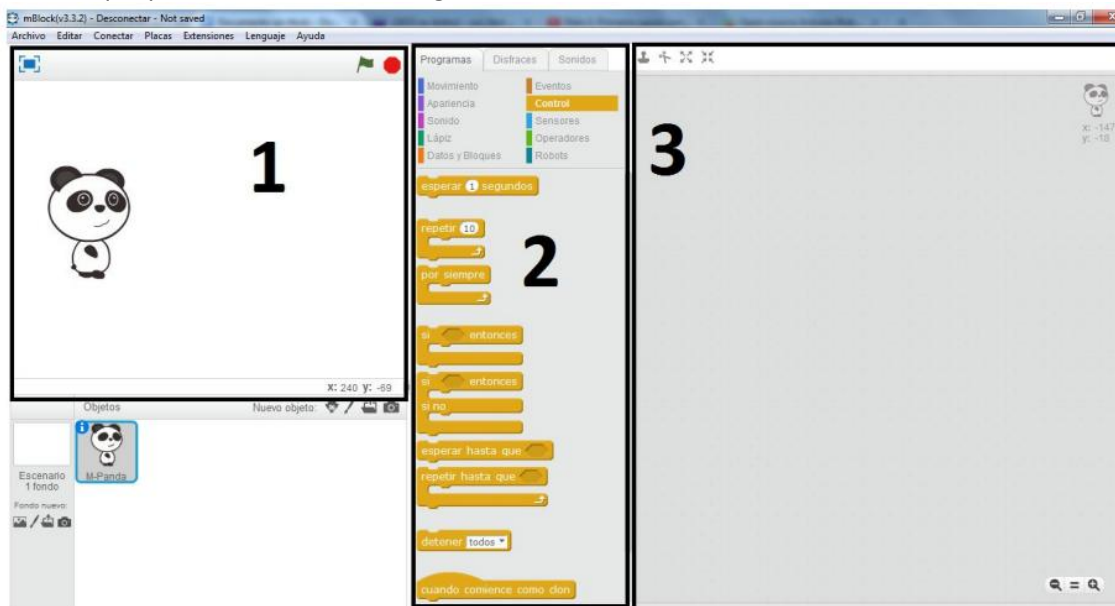
mBlock es un software fácil de aprender y usar, ya que está pensado para niños. Su diseño visual es parecido a un puzzle, donde el usuario juntando bloques (también llamadas piezas) para construir un programa. Y, como en un puzzle, unas encajan entre ellas y otras no, por lo cual es fácil identificar cuando se está haciendo algo bien o mal.

La variedad de bloques disponibles en mBlock permiten programar una amplia posibilidad de comportamientos.

Este tipo de programación se llama programación visual (por bloques) y es muy distinta a la programación por código, la cual es la típica programación escrita.

mBlock está basado, o es una variante de un famoso software libre llamado Scratch 2.0. El cuál está orientado a la educación y es muy conocido en todo el mundo.

Scratch está pensado originalmente para programar videojuegos. Mientras que mBlock está pensado para programar robots. Aunque en mBlock se pueden programar las dos cosas: videojuegos y robots. La interface que presenta mBlock es la siguiente:



Del lado izquierdo, está “el escenario” (1) del videojuego; en el centro (2) están las categorías de bloques de programación (las piezas con las que se construye el programa del videojuego o del robot); y del lado derecho está el espacio en donde se “arma/construye” el programa, dónde se colocan las piezas/bloques.

LOS BLOQUES

En la ventana del centro (2) se encuentran los bloques de programación, agrupados por sus distintas funcionalidades, donde cada funcionalidad está asociada a un color diferente.

Vemos que “movimientos” está asociado al azul, “apariencia” al color lila, sonido al magenta, lápiz al verde mar, bloques y listas al naranja, eventos al marrón, control al naranja melocotón, sensores al azul marino, operadores al verde y robots al azul oscuro.

Estas funcionalidades se pueden dividir en dos grandes grupos.

- **Bloques para video juegos.**

Para programar un videojuego se utilizan las siguientes categorías de bloques: Movimiento, Apariencia, Sonido, Lápiz, Sensores, Eventos, Control, Operadores, Datos y Bloques.

- **Bloques para robot.**

Para programar un robot se utilizan las siguientes categorías de bloques: Robots, Eventos, Control, Operadores, Datos y Bloques.

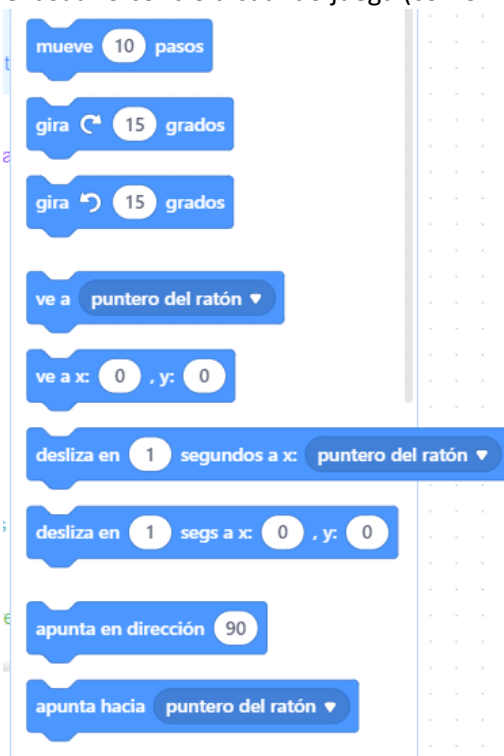
Categoría	Notas	Categoría	Notas
Movimiento	Mueve objetos y cambia ángulos	Eventos	Contiene activadores de eventos situado al comienzo de cada grupo de instrucciones
Aparencia	Controla el aspecto visual del objeto, añade bocadillos de habla o pensamiento, cambia el fondo, ampliar o reducir	Control	Contiene los bloques de lógica de programación como los bucles, condiciones, repeticiones..
Sonido	Reproduce ficheros de audio y secuencias programables	Sensores	Los objetos pueden interactuar con el ambiente que ha creado el usuario
Lápiz	Control del ancho, color e intensidad del lápiz	Operadores	Operadores matemáticos, generador aleatorio de números, comparadores..
Datos y Bloques	Creación de variables, bloques y listas	Robots	Bloques para controlar el robot, usar sensores, girar motores...

DESCRIPCIÓN DE CATEGORÍAS DE BLOQUES

A continuación, se describen las diferentes categorías de bloques disponibles en mBlock. La mayoría sólo sirven para programar videojuegos, la categoría de Robots sólo sirve para programar robots, y hay 3 categorías que comparten los videojuegos y robots (Control, Eventos, Datos y Bloques). Cuando se describen los bloques para los videojuegos se menciona al Oso Panda, el Oso Panda es el “avatar” que predefine para el videojuego mBlock, o sea el “personaje” que el usuario controla cuando juega (como Mario de Mario Bros). El Oso Panda o “avatar” sólo está disponible cuando se programa un videojuego, cuando se programa un robot, no es posible pre-visualizar el comportamiento del robot en el Oso Panda. Nota: El Oso Panda es nuestro ejemplo, pero también se puede sustituir por un Gato o cualquier “avatar” que un usuario diseñe.

Bloques de Movimiento:

Con los bloques azules podemos hacer que el Oso Panda se desplace hacia delante o hacia atrás la distancia que queramos, que rote sobre el mismo, que cambie de dirección, moverlo a las coordenadas deseadas, etc.



Bloques de apariencia (color lila/morado):

Con los bloques lilas se puede hacer que el fondo del “escenario” (1) cambie de color o que el Oso Panda cambie de color, añadir un nuevo personaje, cambiar partes del cuerpo de los personajes, hacer que hablen y muestren un mensaje, cambiar tamaños, etc.



Bloques sonido (color magenta):

Con los bloques magentas se realiza todo tipo de sonidos y se puede controlar el volumen de ellos.

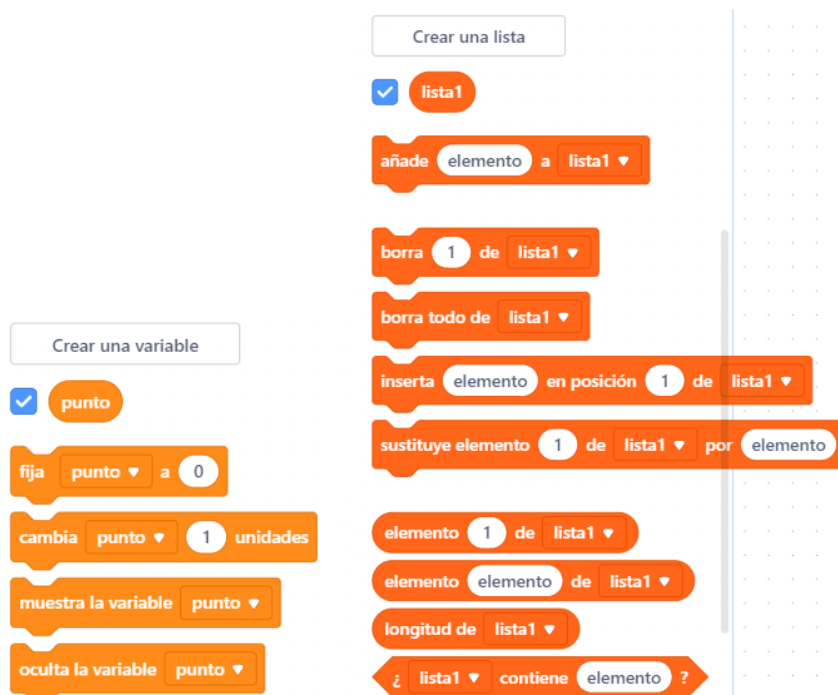


Bloques de Lápiz (color verde mar):

Con los bloques verde mar el “escenario” (1) se convierte en una pantalla interactiva tipo Paint y se puede pintar el escenario como se quiera.

Datos y Bloques (color naranja):

Con los bloques naranja se crean bloques de variables, de listas de datos y bloques dentro del programa. Esto se utiliza para realizar programas más complejos y completos:



¿Qué es una variable?

Una variable es un espacio de memoria en nuestro programa que utilizamos cuando queremos almacenar un valor y trabajar con éste durante la ejecución del programa. Explicado con un ejemplo ilustrativo, si queremos hacer un videojuego donde se tenga 3 vidas, podemos crear una nueva variable llamada Vida y hacer que cada vez que pase algo decrezca el valor en 1.

¿Qué usos podemos darle a una variable?

- Almacenar el número de preguntas acertadas en un juego de preguntas.
- Almacenar el total de puntos que tenemos en un juego de cartas.
- Almacenar el número de repeticiones que llevamos en una iteración.

¿Qué es una lista?

Una lista es lo mismo que una variable, pero en vez de almacenar sólo un valor, podemos almacenar tantos valores como queramos.

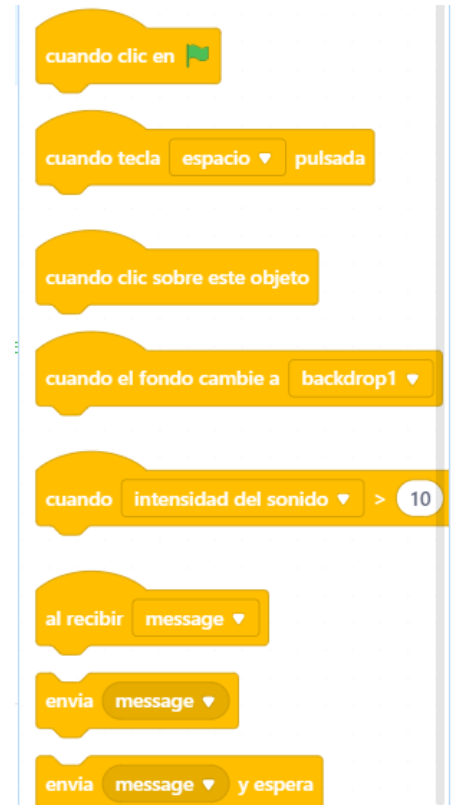
¿Qué usos podemos darle a una variable?

- Almacenar las respuestas en un juego de preguntas.
- Almacenar el total de puntos que hemos conseguido en cada uno de los niveles de un juego.

- Almacenar una lista de palabras válidas.

Bloques eventos (color marrón):

Los bloques marrón sirven para activar el inicio de cualquier programa que hayamos hecho utilizando input externos (como el ratón o las teclas del ordenador) o inputs internos como envío de mensajes del propio programa a otro. Harían la función del botón play en un reproductor de música, cuando lo activamos, empieza todo a funcionar.



Bloque Control (color melocotón):

Con los bloques de melocotón se realiza la lógica de programación, como los bucles, que una acción se repita varias veces, que espere un tiempo para hacer la siguiente acción o indicar que el programa realice una acción si se cumplen ciertas condiciones y si no se cumplen se realice otra. Con la lógica de programación definimos las reglas del mundo del videojuego o las del funcionamiento del robot, como por ejemplo si queremos que se pare o gire cuando nuestro robot o el avatar detecte un obstáculo en su camino.



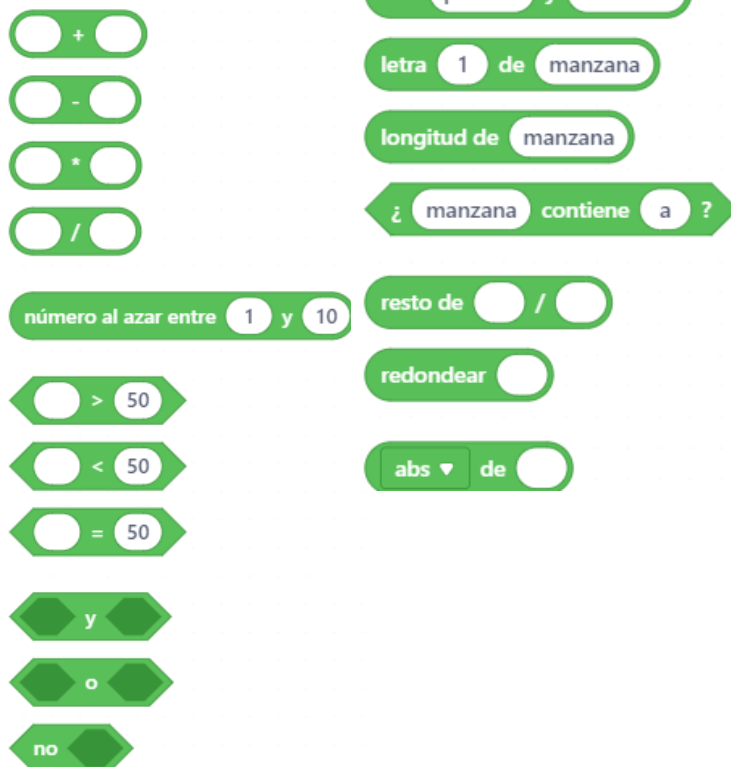
Bloque Sensores (color azul marino):

Con los bloques azul marino permite detectar si el Oso Panda realiza las siguientes acciones: toca otro Objeto, tocar un color determinado, distancia entre un objeto y el oso. También permite detectar cuando un color determinado toca otro color, la entrada de variables por medio del teclado, si se está presionando alguna tecla del ratón. Verificando todas estas posibles acciones, se puede hacer que el Oso salte, pare, retroceda, que el juego acabe, que sume puntos, que cambie la pantalla...



Bloques Operadores (color verde):

Con los bloques verdes se realizan comparaciones de valores, sumas, restas, contadores, uniones de palabras o de números, etc. Por ejemplo, para definir la condición para que el robot u Oso panda se pare a una cierta distancia de un obstáculo, se tendrá que usar el bloque de comparación "menor que". Estos bloques, combinándolos con los de sensores, nos permiten saber si estamos a una distancia menor o mayor de otro objeto, y así poder hacer una u otra acción.

**Bloques Robots (azul fuerte):**

Los bloques Robots, de color azul fuerte, es un grupo de bloques exclusivo, creado por Makeblock tanto para las placas propias como la mCore, la Orion como para las placas Arduinos, las cuales son el "cerebro" de los robots Makeblock.

Este grupo de bloques no existe en el programa estándar de Scratch.

A continuación, hacemos una breve descripción de los bloques Robot para el mBot (robot de Makeblock). Makeblock amplía y actualiza este grupo a medida que introduce nuevos sensores en sus robots.

Bloque mBot Program

Es básicamente el bloque que envía el mensaje de "Ejecutar el programa cargado en el robot" cuando se enciende. Sustituye al bloque de banderita verde cuando se programa un videojuego. Gracias a este bloque, una vez pasado el programa, el robot puede trabajar sin estar conectado al ordenador vía cable o bluetooth.

Para cargar un programa en la placa Arduino, el mBot debe estar previamente conectado al ordenador mediante un cable USB, después se hace clic con el botón derecho sobre el bloque mBot Program y se selecciona "Upload to Arduino".

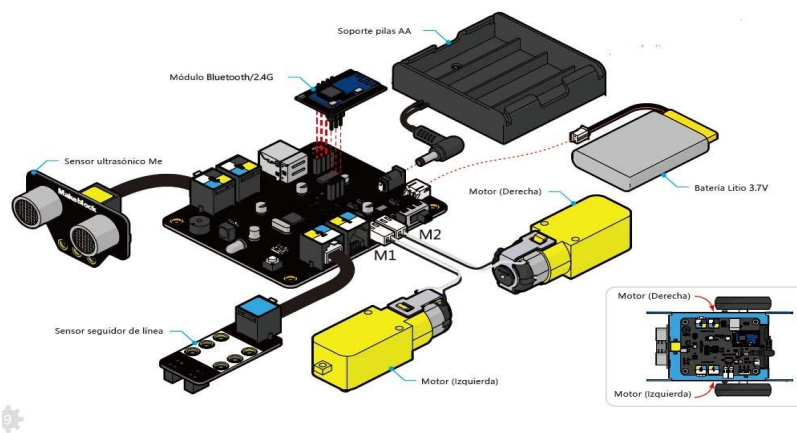
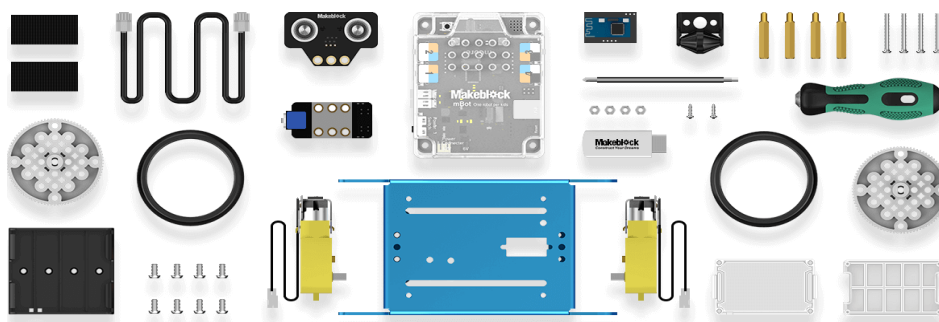
MÓDULO 2

HARDWARE DE MBOT MOTORES

Identificación de elementos de mBot

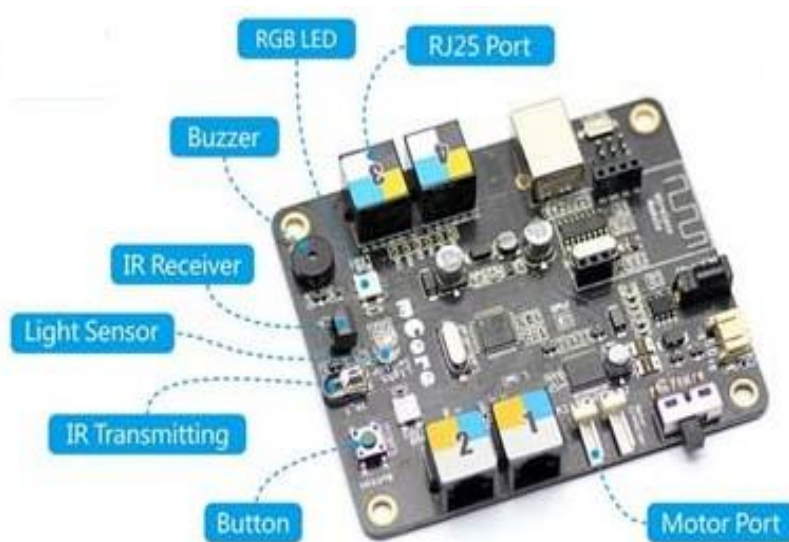
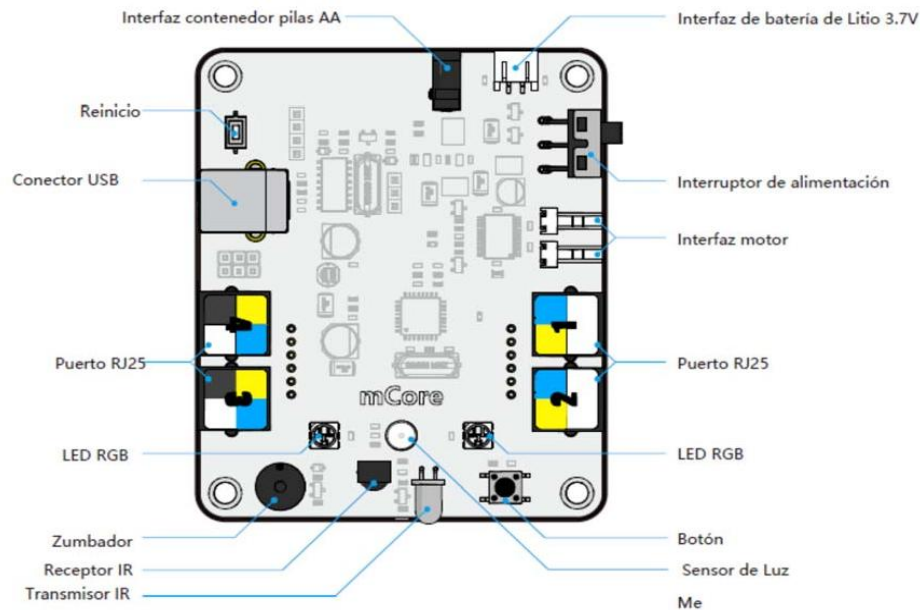
EJERCICIO N°1: Identificando todos los componentes del robot mBot

Intenta identificar todos los componentes del robot, además de saber para qué sirve cada uno de ellos. Puedes usar la imagen inferior para localizarlos dentro de la caja.



Placa mCore

El robot mBot utiliza la placa mCore que puede verse en la imagen inferior. La placa, con un microcontrolador ATmega238, dispone de 4 puertos con conexiones RJ25 para conectar sensores y dos puertos para conectar motores. Además, mCore integra un interruptor de encendido, un botón, dos LEDs RGB, dos LEDs normales, un buzzer, un sensor de luminosidad y un sensor de infrarrojos receptor-emisor.



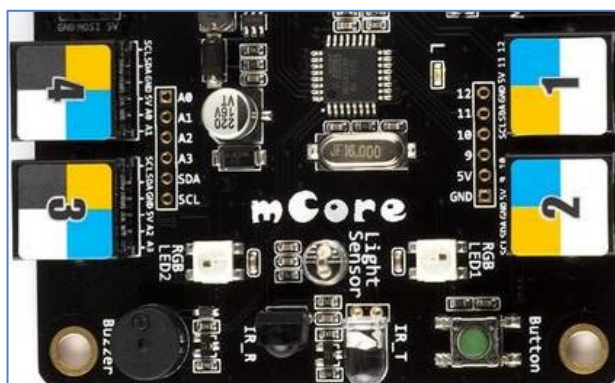
El chasis cuenta con una ranura superior para organizar por donde pasarán los cables de conexión. Además, cuenta con brazos delanteros que evitan impactos de choque sobre los sensores y cuentan con ranuras para la extensión de piezas.

La conexión de cables y posición de los elementos del robot es bastante intuitiva, para las conexiones además se tiene un código de colores que clasifica a los sensores con sus puertos compatibles en la tarjeta. Además, se utilizan cables recubiertos con goma o similares, para aumentar su resistencia

La placa mCore del mBot puede programarse utilizando diferentes lenguajes de programación. De hecho, no es más que una Arduino Uno y, por lo tanto, puede ser programada con Processing. Esto hace que podamos sacarle más rendimiento a la placa ya que, en teoría, tenemos un dos en uno. Aunque en este apartado usaremos muchas veces scratch para programar diferentes módulos, esos mismos

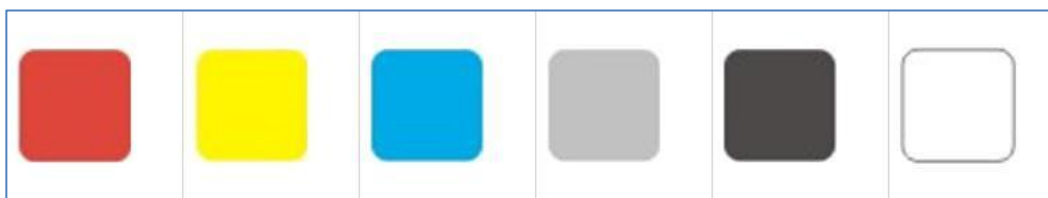
componentes electrónicos podrían ser programados a través del IDE de Arduino y, a veces, usaremos esta posibilidad.

Los módulos que pretendemos conectar a la placa presentan y vienen clasificados por su color ID. Ese color debe corresponder con el color del puerto al cual pretendemos conectarlo. Por ejemplo, en la siguiente imagen vemos que el puerto 2 dispone de tres colores: amarillo, azul y blanco. Pues bien, a él podremos conectar cualquier módulo cuyo RJ25 disponga de, como mínimo, alguno de esos colores. Si el ID del módulo es negro, no podríamos conectarlo al puerto 2 (ni al 1), pero si al 3 o 4.



4 puertos de mCore

Los colores ID que podemos encontrarnos en los puertos de las diferentes placas de Makeblock son: Rojo (motores), Amarillo (interface digital), Azul (interface digital dual), Gris (Puerto serie, bluetooth), Negro (interface analógica y dual) y Blanco (Puerto I²C).



EJERCICIO N°2: Montando el robot mBot

Ensamblar los elementos que componen el robot y una vez montado, probar los elementos que se puedan individualmente.

Materiales:

Todos los materiales incluidos en la caja.

Utilizar la guía de montaje incluida.

Página que explica paso a paso mediante fotos como se monta este robot:

<http://codigo21.educacion.navarra.es/recursos/montaje-de-un-mbot/>



Mover el robot mBot

EJERCICIO N°3: Manejo del robot sin usar las apps

Nada más montado el robot es posible usarlo de varias formas sin necesidad de usar ninguna app.

Se puede usar de dos formas, mediante un pulsador que incorpora en la parte superior o usando el mando a distancia:

A. Pulsador en la parte frontal superior izquierda del robot

1. Una vez encendido y si se presiona este pulsador el robot comenzará a moverse solo y a esquivar obstáculos.
2. Si se presiona una segunda vez el pulsador activará el programa de sigue líneas.
3. Con la tercera pulsación el robot se para.



B. El mando de infrarrojos permite que el robot funcione de tres formas diferentes

4. Si se pulsa la tecla A se puede manejar el robot por control remoto usando las cuatro flechas del mando, una por cada dirección.
5. Al pulsar la tecla B se activa volverá a mover solo y a esquivar de nuevo obstáculos
6. Para activar el programa sigue-líneas se presiona la tecla C del mando

Este ejercicio consiste en probar cada una de las opciones de control del robot descritas en la parte superior.

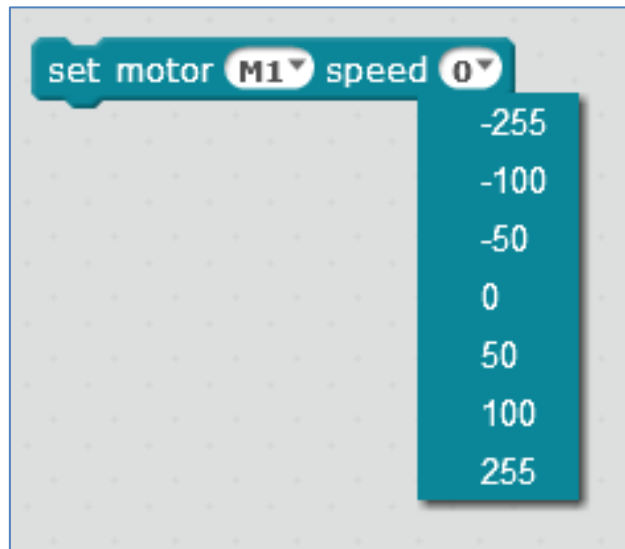


MOTORES

El robot mBot se compone de dos motores de continua de 6V a 200rpm. En la placa, estos motores se denotan por M1 y M2, pudiendo variar su velocidad, a la hora de programarlos, desde 0 a 255. El valor mínimo, 0, es la velocidad más lenta y 255 es la más rápida. Los valores negativos harán que gire en el otro sentido (rotación inversa).

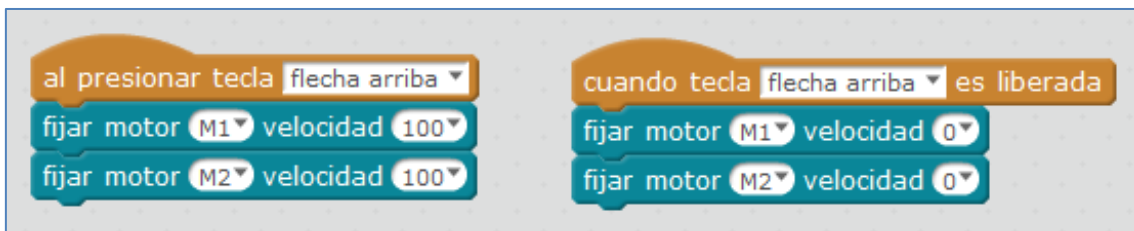


Motor DC 6V/200rpm

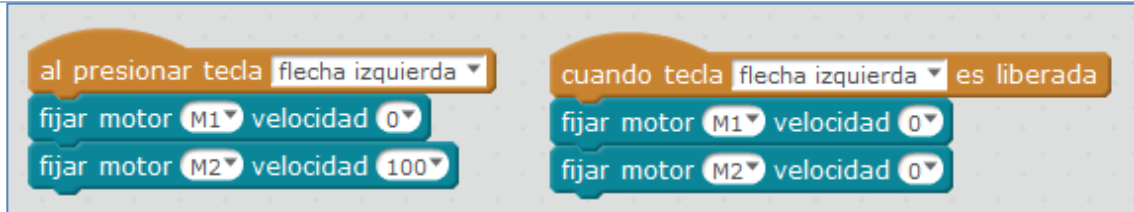


Podemos programar, por ejemplo, el movimiento de los motores a través del teclado del ordenador, al presionar o liberar una tecla. Debemos decidir qué ocurrirá cuando presiono la tecla "X" y qué ocurrirá cuando dejen de presionar la tecla "X".

Por ejemplo: Cuando pulse la tecla "flecha arriba" los dos motores comienzan a funcionar hacia adelante, con una velocidad de 100, pero, cuando dejen de presionar esa tecla, se pararán.



Para que gire hacia la izquierda, debemos poner el motor izquierdo a 0 (M1 o M2, dependiendo de cómo se conectaran²) y el derecho a una velocidad (M1 o M2) por ejemplo de 100. Al dejar de presionar, se debe parar. En el ejemplo que vemos en la siguiente figura, M1 es el motor izquierdo y M2 el derecho:



Se puede mejorar el script anterior haciendo que M1 gire en el sentido contrario, con velocidad -100. El movimiento de control completo en un coche sería:



Script para el movimiento de los motores con el teclado

Con el tiempo y los avatares que vaya pasando el robot, quizás debamos reemplazar alguna pieza de un motor del mBot.

ELECTRÓNICA BÁSICA ORIENTADO A LA ROBÓTICA

Electrónica básica orientada a la robótica

¿Cómo son los robots por dentro?

En el interior de un robot se encuentran componentes electrónicos, el cableado y otros pertenecientes al sistema eléctrico del robot.



¿Qué es la electrónica?

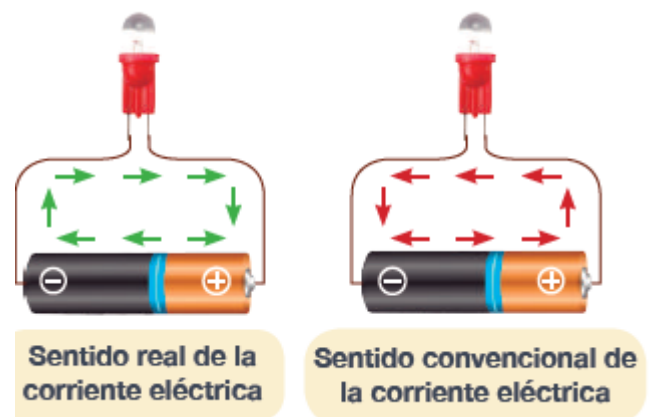
La electrónica es una disciplina que se encarga de estudiar a la electricidad, como se conduce, maneja y transmite en los distintos medios, es decir entre las cosas que vemos.

Su finalidad práctica es la elaboración de circuitos que sirvan para la confección de artefactos eléctricos de utilidad para la sociedad.

La corriente eléctrica y su utilidad en robótica educativa

La corriente eléctrica es el movimiento de los electrones a través de un circuito eléctrico cerrado, desde el terminal negativo al terminal positivo de una batería eléctrica, impulsados por la diferencia de potencial existente.

Para facilitar la comprensión de la electrónica, utilizamos el “sentido convencional” del flujo de electrones. Esto significa que, en lugar de considerar que los electrones se mueven del terminal negativo al positivo de una batería, imaginamos que se desplazan en dirección opuesta, es decir, del terminal positivo al negativo. Esta convención simplifica el análisis de los circuitos y nos permite entender mejor cómo se comportan los componentes electrónicos en un circuito eléctrico cerrado.



AC



DC



Tipos de corriente eléctrica

Existen dos tipos de corriente eléctrica:

- **AC. (Corriente Alterna).** No tiene polaridad, podemos encontrar en la toma de corriente en nuestras casas.
- **DC. (Corriente Continua).** Mantiene siempre fija su polaridad, es decir tiene un positivo y un negativo. En robótica educativa se utiliza este tipo de corriente eléctrica.

Fuentes de energía eléctrica

La fuente de energía eléctrica más usada en los robots son las baterías, estas alimentan a todo el sistema, los robots se alimentan con energía eléctrica, así como nosotros nos alimentamos con comida. Toda batería o fuente de energía eléctrica tiene dos características muy importantes, el voltaje y el amperaje.

Voltaje y tensión

Es como una fuerza que impulsa a los electrones en un circuito cerrado y se mide en voltios (V).

Amperaje o intensidad

Es la cantidad de electrones que fluyen en un determinado tiempo, y se mide en amperios (A).

Conductores y aislantes

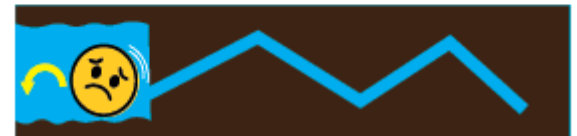
Los conductores son como el camino para los electrones, por ahí circula la electricidad, por ejemplo, los materiales metálicos.

Los aislantes son los que no dejan circular a la electricidad, por ejemplo, el plástico.

Conductores



Aislantes

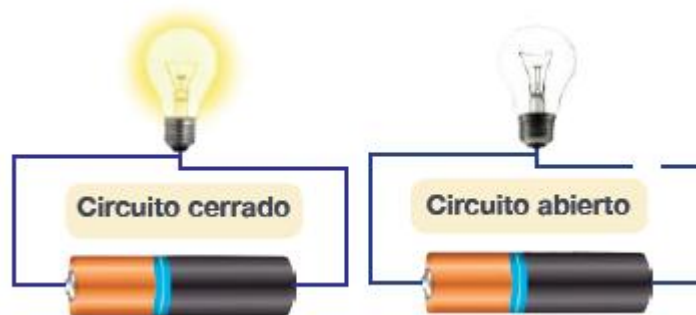


Circuitos

Son el recorrido o camino que sigue la corriente para llegar de un punto a otro.

Existen dos tipos de circuitos:

- Circuito eléctrico abierto; es cuando existe la interrupción del flujo de corriente eléctrica.
- Circuito eléctrico cerrado; es cuando existe un flujo de corriente eléctrica a través del circuito, no obstante, hay tres requisitos importantes para que exista este flujo de corriente eléctrica, una fuente de energía o batería, un camino para recorrer o cables y un consumidor o algo que consuma energía eléctrica.



MÓDULO 3

SENSORES

Alternativa condicional

Las condiciones en las cuales se ejecutan los programas suelen ser diversas. Por lo tanto, al programar, debemos evaluar los distintos escenarios que pueden presentarse y las acciones adecuadas a realizar en cada caso. En este módulo se presenta la alternativa condicional, que es una herramienta de los lenguajes de programación para que un programa se comporte de uno u otro modo de acuerdo con ciertas condiciones de los datos.

Permite, por lo tanto, construir programas versátiles que funcionan en distintos escenarios. Cuando se escriben algoritmos o programas, podemos encontrarnos frente a la necesidad de programar diferentes alternativas a partir de una condición. En determinados escenarios, estas condiciones pueden tomar valores verdaderos o falsos.

Ejemplo: Si hay música, entonces nos movemos

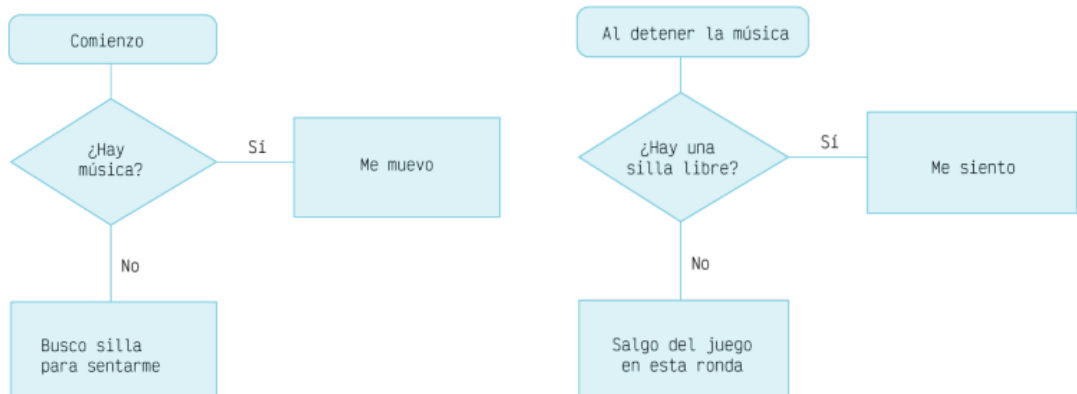
Estas estructuras están formadas por una condición y una acción, la cual solo se realiza si la condición es verdadera, es decir, permite realizar una determinada acción según una condición.

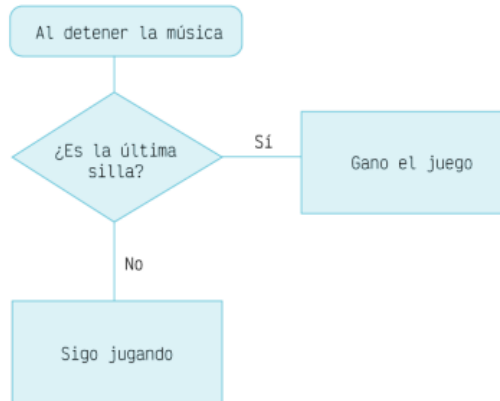
Analicemos el juego de la silla repasando su dinámica para después generar las leyes. Seguramente, muchos lo conocen ya, sin embargo, se sugiere poner en común las reglas. Compartimos a continuación las reglas.

- Cuando comienza a sonar la música, todos deberán moverse y girar alrededor del círculo de las sillas siguiendo el ritmo de la canción.
- Cuando la música se detenga, cada uno deberá sentarse en una silla.
- El que se quede sin silla, quedará fuera de esta ronda de juego.
- Se quita una silla y se reanuda el juego.
- Se repite el mismo procedimiento hasta que quede solo una silla y dos participantes.
- El último que quede sentado será el ganador del juego.

Para conversar

- ¿Qué hacemos cuando suena la música?
- ¿Y cuándo se detiene?
- ¿Qué pasa si no quedan sillas para sentarse cuando se detiene la música?
- ¿Cuándo termina el juego?
- ¿Cuál es la relación entre las veces que se detiene la música y la cantidad de participantes?





Sensor de ultrasonidos

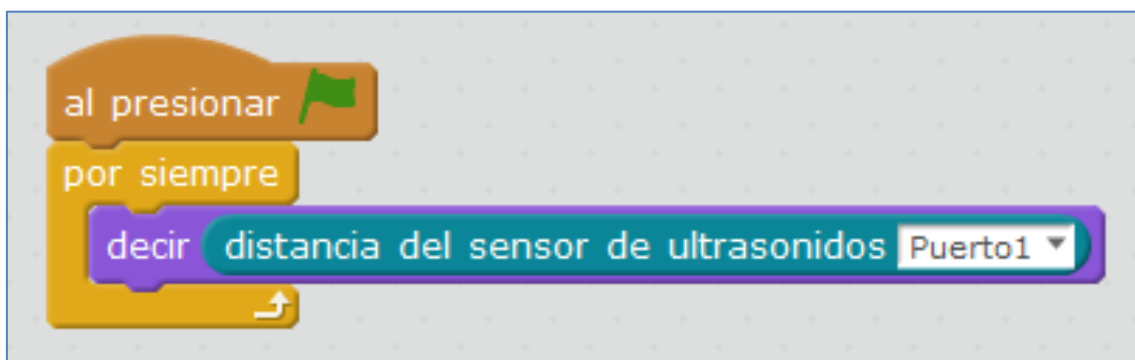
El sensor de ultrasonidos o distancia son detectores de proximidad que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar. Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración.

Un módulo de ultrasonidos nos proporciona un dato numérico que se corresponde con la distancia entre el sensor y cualquier objeto que está en frente de él. Por lo tanto, se utiliza para medir distancias, logrando detectar objetos que se encuentran a 3 o 4cm del sensor. Su color ID es amarillo y eso significa que puedo conectarlo a cualquiera de los cuatro puertos de una placa mCore del mBot.

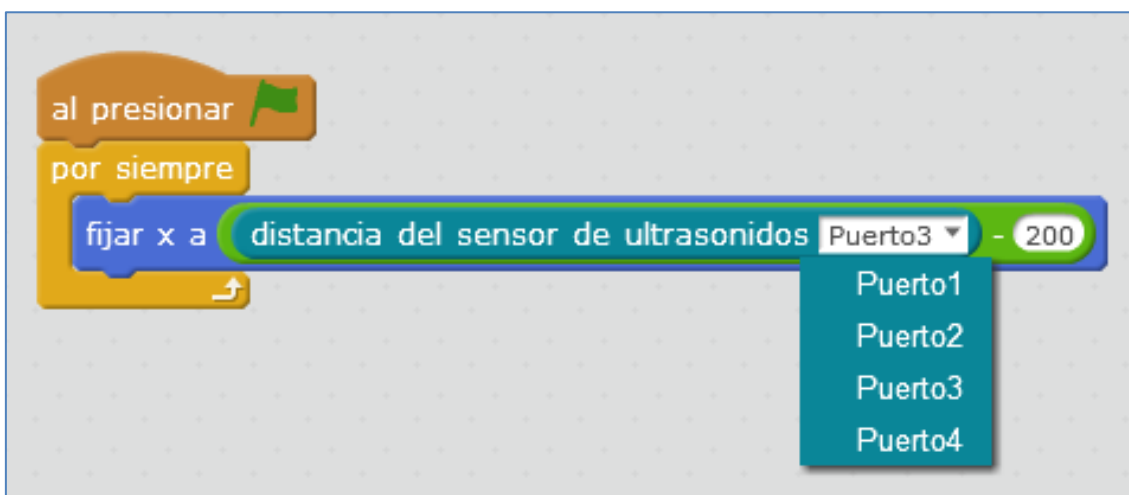


Módulo ultrasonidos

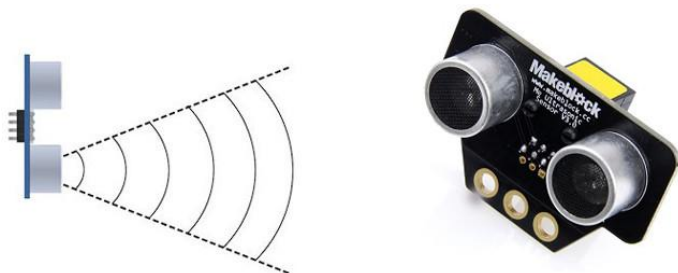
Supongamos que lo conectamos al puerto 1 de la placa mCore del robot. El siguiente script, nos mostrará la distancia que mide en el escenario del mBlock:



También podemos controlar el movimiento del ratón con el sensor de ultrasonidos. Una forma sería con el siguiente script:

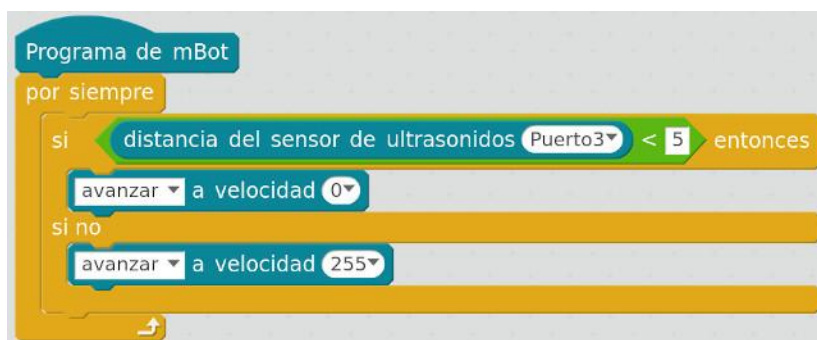


NOTA: Ojo, hay que ver en qué puerto de los 4 tenemos conectado nuestro sensor de ultrasonidos. El puerto típico para el mBot suele ser el puerto 3.



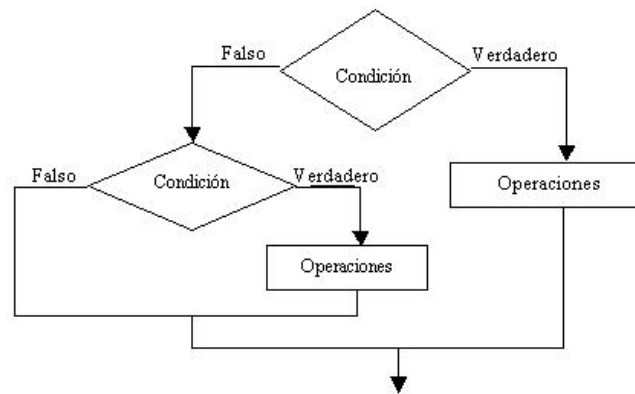
El funcionamiento del sensor es muy sencillo; se utiliza un emisor para emitir una onda de sonido de alta frecuencia (ultrasonido) y un receptor para escuchar la onda reflejada. En mBlock solamente tenemos que utilizar el bloque que nos devuelve la longitud a la cual se encuentra de un obstáculo. A partir de esa distancia podemos actuar según nuestro objetivo.

En este apartado se va a programar un sencillo código encargado de detectar un obstáculo y detenerse ante él. Para ello, utilizaremos una condición encargada de comparar el valor obtenido del sensor de ultrasonidos, y en caso de ser mayor que 5 centímetros, parará los motores (izquierdo - M1 y derecho - M2). En caso contrario, los motores funcionarán al máximo permitido en mBlock, que es el valor de 255.



Estructuras condicionales anidadas

Decimos que una estructura condicional es anidada cuando por la rama del verdadero o el falso de una estructura condicional hay otra estructura condicional.



El diagrama de flujo que se presenta contiene dos estructuras condicionales. La principal se trata de una estructura condicional compuesta y la segunda es una estructura condicional simple y está contenida por la rama del falso de la primera estructura.

Es común que se presenten estructuras condicionales anidadas aún más complejas

Problema:

Confeccionar un diagrama que indique qué ropa usarías dependiendo a la temperatura, con tres posibilidades:

Si la temperatura está calurosa indicar la ropa a utilizar.

Si la temperatura está templada indicar la ropa a utilizar.

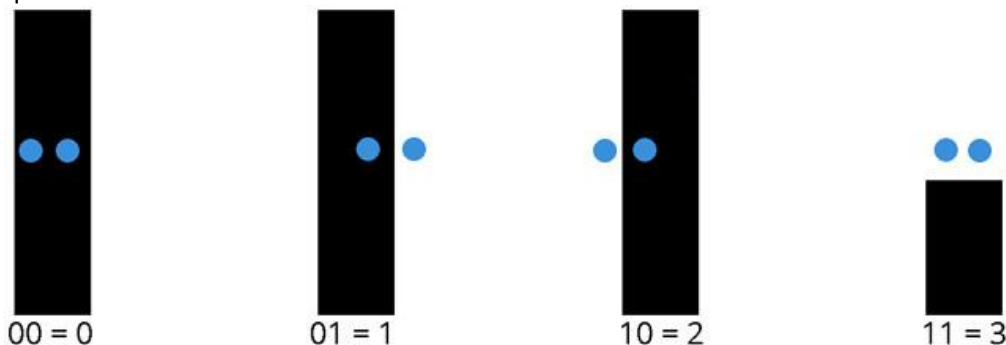
Si la temperatura está fría indicar que usarías.

Sensor infrarrojo (sigue-líneas)

El sigue-líneas o sensor de infrarrojos (IR) es un sensor de medición de distancia que se basa en un sistema de emisión y recepción de radiación lumínica en el espectro de los infrarrojos. Dicho de otro modo, es un sensor con una fuente de luz (diodo emisor) y detector (fototransistor) integrados en un mismo encapsulado. La detección del objeto se consigue por la reflexión (o no) del haz infrarrojo sobre la superficie del objeto. Este sensor se utiliza habitualmente para detectar el color de un objeto, en nuestro caso, distinguir blanco y negro en aplicaciones para el seguimiento de línea.



El funcionamiento del sensor es el siguiente; si el sensor del robot mBot está sobre la línea negra, los reflejos del sensor son absorbidos por la misma con lo cual el receptor del sensor no recibe ningún valor, o lo que es lo mismo, dando el valor 0. Del mismo modo, si uno de los dos sensores del robot queda fuera de la línea, se producirá el reflejo indicando los valores 1 o 2 respectivamente. Por último, si ambos sensores quedan fuera de la línea se mostrará el valor 3.



- Valor 0: mBot tiene los dos detectores IR sobre la línea.
- Valor 1: mBot tiene el detector IR de la derecha fuera de la línea.
- Valor 2: mBot tiene el detector IR de la izquierda fuera de la línea.
- Valor 3: mBot tiene ambos detectores IR fuera de la línea.

Es muy importante conocer las conexiones donde hemos conectado nuestro robot para que no realice movimientos inesperados. En nuestro caso hemos seguido las conexiones que vienen por defecto en el manual de construcción del robot.



```
Programa de mBot
por siempre
si sigue-líneas Puerto2 = 0 entonces
  fijar motor M1 velocidad 100
  fijar motor M2 velocidad 100
si sigue-líneas Puerto2 = 1 entonces
  fijar motor M1 velocidad 0
  fijar motor M2 velocidad 100
si sigue-líneas Puerto2 = 2 entonces
  fijar motor M1 velocidad 100
  fijar motor M2 velocidad 0
si sigue-líneas Puerto2 = 3 entonces
  fijar motor M1 velocidad 0
  fijar motor M2 velocidad 0
```

MÓDULO 4

RECURSOS Y MEDIOS TÉCNICOS PARA LA PRODUCCIÓN

Objetivos:

- Reconocer los diferentes medios técnicos y su contribución en el accionar tecnológico.

Máquinas y mecanismos

Máquinas y mecanismos. Tipos

El hombre a lo largo de la historia ha inventado una serie de dispositivos o artugios llamados máquinas que le facilitan y, en muchos casos, posibilitan la realización de una tarea.



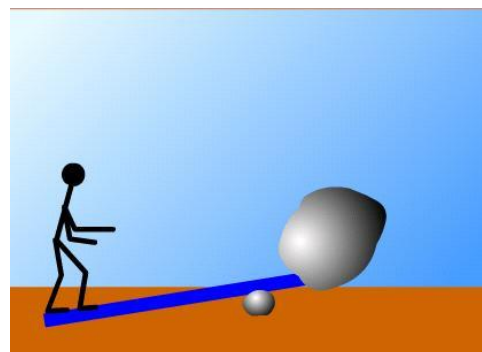
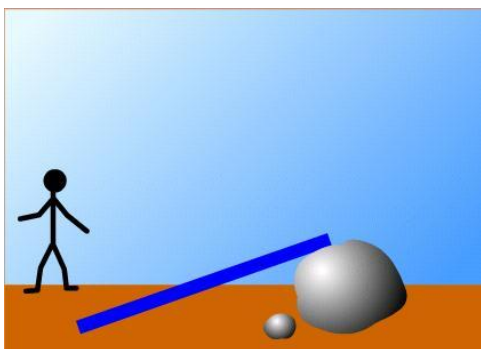
Una **máquina** es el conjunto de elementos fijos y/o móviles, utilizados por el hombre, y que permiten reducir el esfuerzo para realizar un trabajo (o hacerlo más cómodo o reducir el tiempo necesario).

Fragua hidráulica. La fuerza del agua movía el martillo, facilitando la labor para elaborar todo tipo de herramienta



Las máquinas son dispositivos inventados por el hombre para ayudarlo a realizar trabajos con un menor esfuerzo.

Para mover una piedra de 40 Kg, un niño de 20 Kg lo podrá hacer con ayuda de una **palanca**.



La palanca hace la otra parte del trabajo por nosotros.

Cuando más larga sea, mayor será el peso que podamos levantar con ella. Solo tendremos que situar bien el punto de apoyo.

La parte del trabajo que una máquina hace por nosotros es lo que se denomina.

VENTAJA MECANICA

Para realizar grandes esfuerzos necesitaremos máquinas que tengan una ventaja mecánica también grande

Prácticamente cualquier objeto puede llegar a convertirse en una máquina, sólo hay que darle la utilidad adecuada. Por ejemplo, una cuesta natural no es, en principio, una máquina, pero se convierte en ella cuando el ser humano la usa para elevar objetos con un menor esfuerzo (es más fácil subir objetos por una cuesta que elevarlos a pulso). Lo mismo sucede con un simple palo que nos encontramos tirado en el suelo, si lo usamos para mover algún objeto a modo de palanca ya lo hemos convertido en una máquina.

Las máquinas suelen clasificarse atendiendo a su complejidad en máquinas simples y máquinas compuestas:

Máquinas simples: realizan su trabajo en un sólo paso o etapa. Por ejemplo las tijeras donde sólo debemos juntar nuestros dedos. Básicamente son tres: la **palanca**, la **rueda** y el **plano inclinado**. Muchas de estas máquinas son conocidas desde la antigüedad y han ido evolucionando hasta nuestros días.



En el plano inclinado el esfuerzo será tanto menor cuanto más larga sea la rampa. Del plano inclinado se derivan muchas otras máquinas como el hacha, los tornillos, la cuña.

Máquinas complejas: realizan el trabajo encadenando distintos pasos o etapas. Por ejemplo, un cortaúñas realiza su trabajo en dos pasos: una palanca le transmite la fuerza a otra, la cual se encarga de apretar los extremos en forma de cuña.

Mientras que las estructuras (partes fijas) de las máquinas soportan fuerzas de un modo estático (es decir, sin moverse), los mecanismos (partes móviles) permiten el movimiento de los objetos.

Los **mecanismos** son los elementos de una máquina destinados a transmitir y transformar las fuerzas y movimientos desde un elemento motriz, llamado **motor** a un elemento receptor; permitiendo al ser humano realizar trabajos con mayor comodidad y/o, menor esfuerzo (o en menor tiempo).

Las máquinas simples son:

La palanca



La polea



El plano inclinado



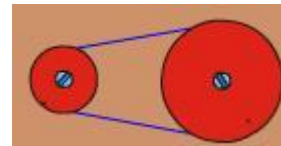
El torno



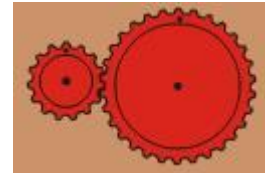
Estas máquinas pueden mezclarse para formar otros mecanismos más complejos.

Para la fabricación de mecanismos complejos usamos:

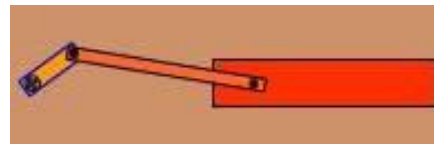
Poleas para transmitir movimiento



Ruedas dentadas



Ruedas excéntricas y bielas



Manivelas y bielas

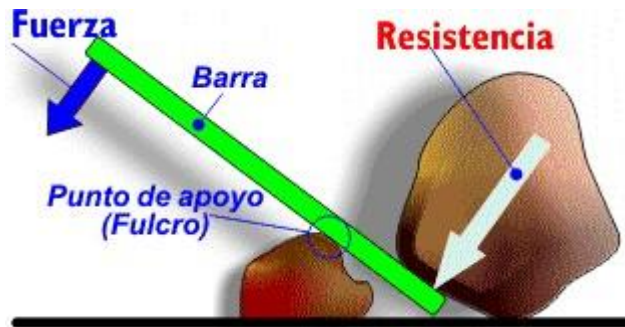
Levas



LA PALANCA

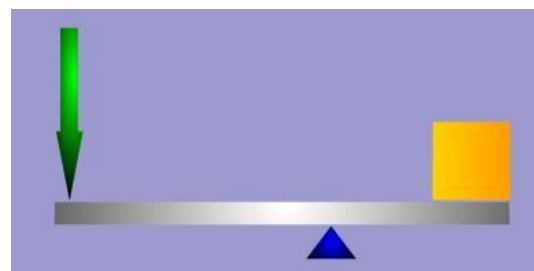
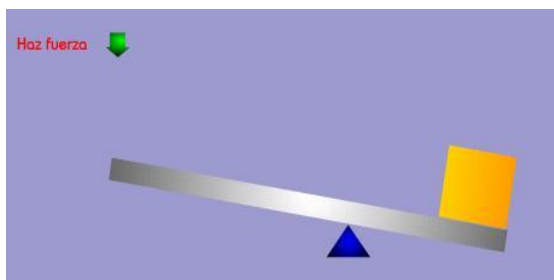
Las palancas son objetos rígidos que giran entorno un **punto de apoyo o fulcro**. En un punto de la barra se aplica una **fuerza o potencia (F)** con el fin de vencer una **resistencia (R)**. Al realizar un movimiento lineal de bajada en un extremo de la palanca, el otro extremo experimenta un movimiento lineal de subida. Por tanto, la palanca nos sirve para transmitir fuerza o movimiento lineal

La palanca es una máquina simple que consiste en una barra rígida que puede oscilar sobre un eje o punto de apoyo.



Al realizar una fuerza en uno de los extremos de la palanca, de modo que baje, el otro extremo sube. Esto quiere decir que la palanca nos sirve para transmitir el movimiento.

La palanca también nos sirve para obtener una ganancia mecánica, es decir, realizando un esfuerzo sobre uno de los extremos de la palanca podemos mover un gran peso que se encuentre en el otro extremo.



La palanca se encuentra en equilibrio cuando el producto de la fuerza (**F**), por su distancia al punto de apoyo (**d**) es igual al producto de la resistencia (**R**) por su distancia al punto de apoyo (**r**). Esta es la denominada **ley de la palanca**, que matemáticamente se expresa como:

$$F \cdot d = R \cdot r$$

Donde:

F: Fuerza o potencia.

d: Brazo de la fuerza, es la distancia desde el punto donde se ejerce la fuerza al punto de apoyo.

R: Resistencia

r: Brazo de la resistencia, es la distancia desde el punto donde se encuentra la resistencia a vencer al punto de apoyo.

Hay tres tipos (géneros o grados) de palanca según se sitúen la fuerza, la resistencia y el punto de apoyo:

En una palanca existen tres elementos característicos:



Potencia: es la fuerza que realizamos.



Resistencia: es la fuerza o carga que se quiere vencer.



Punto de apoyo o Fulcro: es donde se sostiene la barra.



Tipos de palancas

Las palancas se clasifican dependiendo de cómo estén colocados sus elementos fundamentales: **Potencia** – **Resistencia** - **Fulcro**

Podemos encontrar tres tipos de palancas:



Palanca de 1º grado



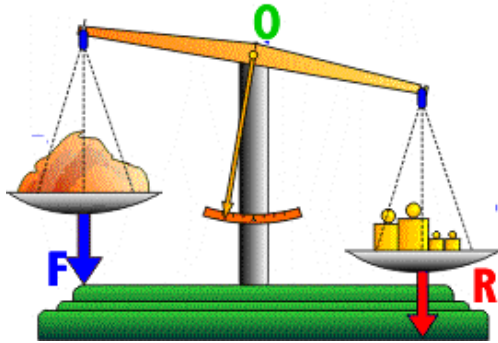
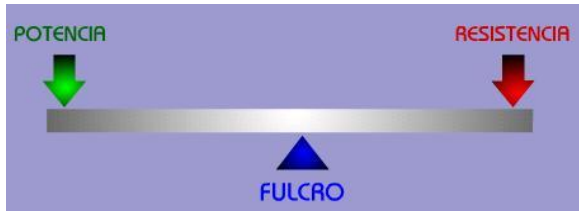
Palanca de 2º grado



Palanca de 3º grado

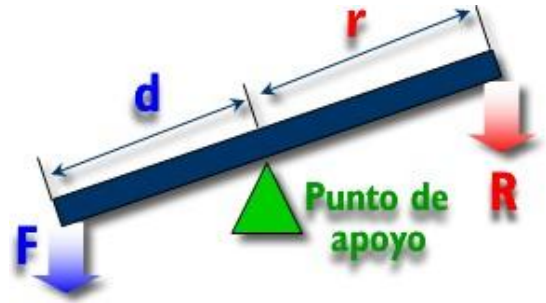
Palanca de 1º grado

Se caracteriza por tener el punto de apoyo entre la **Potencia** y la **Resistencia**, esto no quiere decir que el **Fulcro** esté exactamente en el centro de la barra.



PRIMER GRADO O GÉNERO

El punto de apoyo (O) se encuentra entre la fuerza aplicada (F) y la resistencia (R).

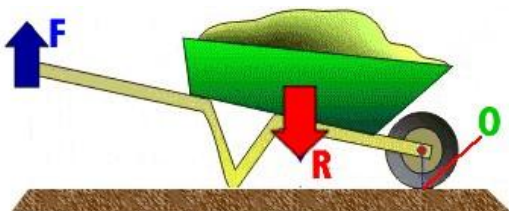


Dependiendo de la colocación del punto de apoyo, la fuerza a aplicar puede ser menor (si $d < r$) o mayor (si $r < d$) que la resistencia.

Ejemplos: Balancín, balanza, tijeras, alicate, martillo (al sacar un clavo), remo de una barca, pinzas de colgar ropa....

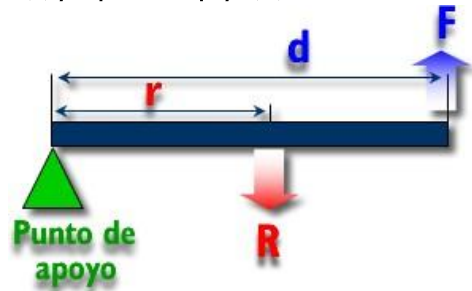
Palanca de 2º grado

Se caracteriza por tener el punto de apoyo en un extremo de la barra, la **Potencia** en el otro extremo y la **Resistencia** en algún punto intermedio.



SEGUNDO GRADO O GÉNERO

La resistencia (R) se encuentra entre la fuerza aplicada (F) y el punto de apoyo (O)

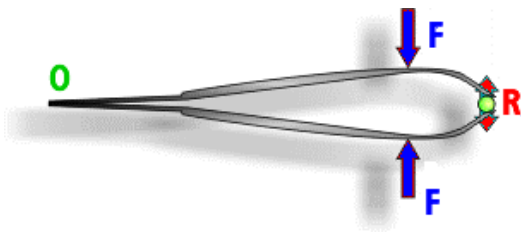
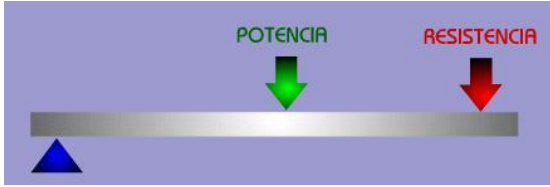


La fuerza a aplicar siempre es menor que la resistencia, ya que $d > r$.

Ejemplos: Carretilla, cascanueces, fuelle, abridor de botellas...

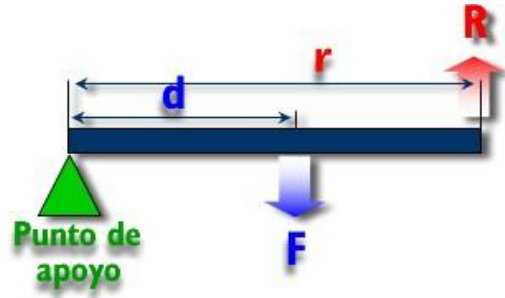
Palanca de 3º grado

Se caracteriza por tener el punto de apoyo en un extremo de la barra, la **Resistencia** en el otro extremo y la **Potencia** en algún punto intermedio.



TERCER GRADO O GÉNERO

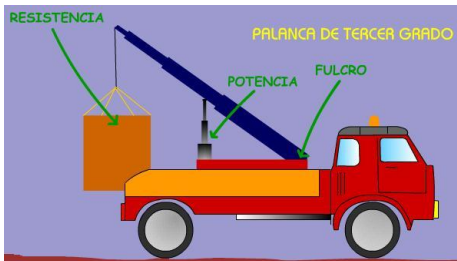
La fuerza a aplicar (F) se encuentra entre la resistencia a vencer (R) y el punto de apoyo (O)

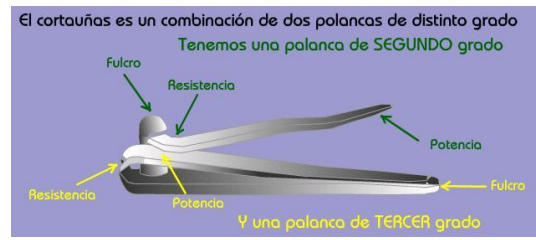


La fuerza a aplicar es siempre mayor que la resistencia, ya que $d < r$.

Ejemplos: caña de pescar, pinzas de depilar, pinzas de hielo, escoba (al barrer), remo de una canoa, banderas, pala de arena

Ejemplos de palancas





LEY DE LA PALANCA

LEY DE LA PALANCA

La facilidad con la que se vence una resistencia depende tanto de la fuerza que se ejerce sobre la palanca como de la posición que ocupa el punto de apoyo.



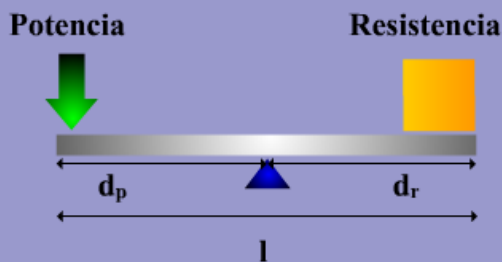
En la primera palanca la fuerza necesaria para vencer la resistencia es mayor que en la segunda palanca.

¿Qué diferencia la primera de la segunda palanca?

Las palancas se diferencian en la posición del punto de apoyo.



Cuanto mayor sea la distancia del punto de apoyo a la potencia menos fuerza tendremos que hacer para vencer la resistencia.



d_p = distancia desde la potencia al fulcro

d_r = distancia desde la resistencia al fulcro

l = longitud de la palanca

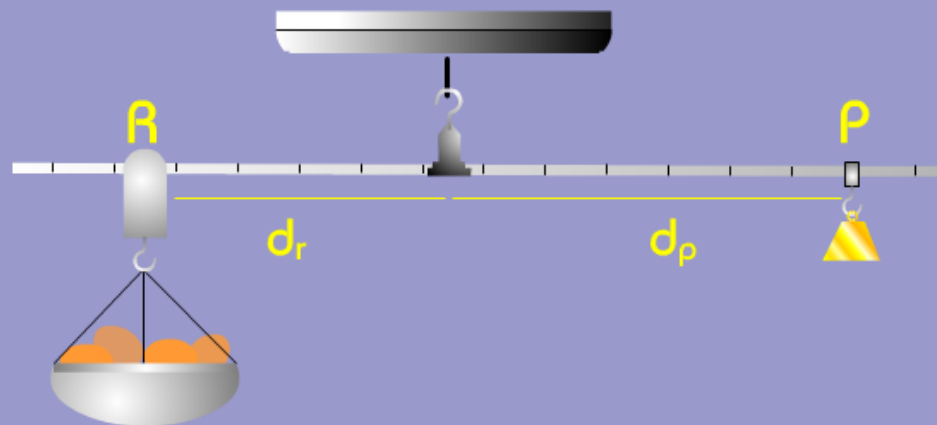
La relación que existe entre la potencia, la resistencia y las distancias de ambas al punto de apoyo se denomina **LEY DE LA PALANCA**

"El producto de la potencia por su distancia al punto de apoyo es igual al producto de la resistencia por su distancia al mismo punto"

Expresada en forma matemática:

$$d_p \cdot P = d_r \cdot R$$

BALANZA ROMANA



Si acortamos un poco el brazo del contrapeso la romana se equilibra. Ahora ya podemos calcular el peso

$$R \cdot d_r = P \cdot d_p$$



LA POLEA

Una polea es una máquina simple, un dispositivo mecánico de tracción, que sirve para transmitir una fuerza. Además, formando conjuntos —aparejos o polipastos— sirve para reducir la magnitud de la fuerza necesaria para mover un peso.

Una **polea** es una rueda ranurada que gira alrededor de un eje. Este se encuentra sujeto a una superficie fija. Por la ranura de la polea se hace pasar una cuerda o cable que permite vencer de forma cómoda una **resistencia** (R) aplicando una **fuerza**. La polea es el punto de apoyo de una cuerda que moviéndose se arrolla sobre ella sin dar una vuelta completa, actuando en uno de sus extremos la resistencia (R) y en otro la fuerza actuante (F) o potencia.

Las poleas simples no son más que una palanca de primer grado, en la que la fuerza F , y la resistencia R se aplican a la misma distancia del eje, que coincide con el radio de la polea. ÉS decir:

$$F \cdot r = R \cdot r$$

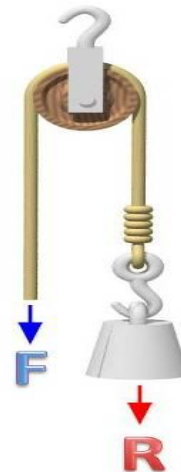
Simplificando tenemos que:

$$F = R$$

Que nos indica que en una polea simple la fuerza que tenemos que aplicar es igual a la resistencia que tenemos que vencer. La ventaja viene del uso de la propia polea que me permite utilizar el cuerpo para ejercer la fuerza F .

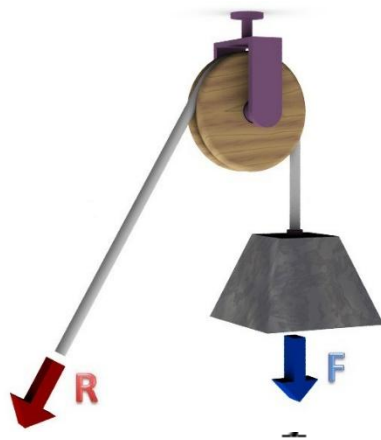


Diferentes tipos de poleas



Polea simple

Tipos de poleas



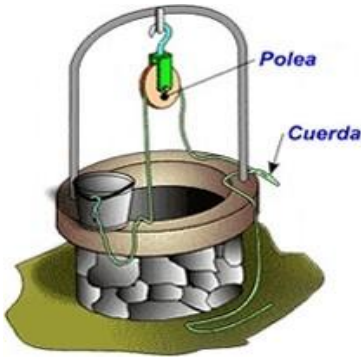
POLEA FIJA

En este caso tenemos una sola polea fija sobre la que se enrolla la cuerda (o cadena) de la que suspende por un lado la carga, que ejerce una fuerza de resistencia R , y del otro lado por donde aplicamos la fuerza F para elevar la carga.

Una polea fija está en equilibrio cuando la fuerza aplicada F es igual a la resistente R que presenta la carga, es decir:

Polea simple fija

$$F = R$$



POLEA FIJA

Se encuentra en equilibrio cuando la fuerza a aplicar (F) es igual a la resistencia (R) que presenta la carga; es decir cuando $F = R$.

El realizar un trabajo con una polea fija no supone un esfuerzo menor, aunque sí más cómodo, cambiando la dirección de la fuerza..



POLEA MOVIL

La polea móvil es un conjunto formado por dos poleas. Una de ellas está fija, mientras la otra puede desplazarse linealmente al subir y bajar la carga. Este tipo de poleas permite elevar cargas con un menor esfuerzo, (con una fuerza aplicada F menor).

Una polea móvil estará en equilibrio cuando se cumple la siguiente igualdad:

$$F = \frac{R}{2}$$

Polea móvil

De esta forma, la fuerza que tenemos aplicar para elevar la carga es la mitad de fuerza resistente de esa carga. Si combinamos varias poleas móviles, la fuerza que necesitamos aplicar disminuirá proporcionalmente al número de poleas móviles del sistema.

POLEA MÓVIL

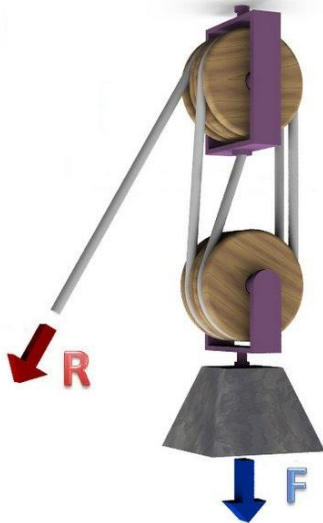
Polea conectada a una cuerda que tiene uno de sus extremos fijo y el otro móvil, de modo que puede moverse linealmente.

La polea móvil se encuentra en equilibrio cuando $F = R/2$; por lo que mediante este sistema la fuerza a realizar para vencer una resistencia se reduce a la mitad. En contrapartida, se necesita tirar del doble de cuerda de la

que habría sido necesaria con una polea fija.

que
la

POLIPASTO



El polipasto es un tipo especial de montaje de poleas fijas y móviles. Consta de un número par de poleas de las que la mitad son fijas, y la otra mitad móviles. Permiten elevar grandes cargas con la aplicación de fuerzas limitadas, por lo que son de gran aplicación en la elevación de grandes cargas (grúas, ascensores, etc).

El polipasto se encuentra en equilibrio cuando se cumple que:

$$F = \frac{R}{2^n}$$

Donde n es el número de poleas móviles

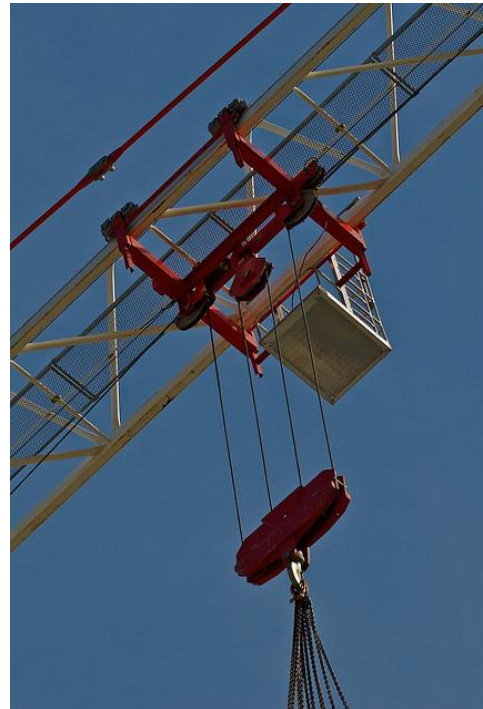
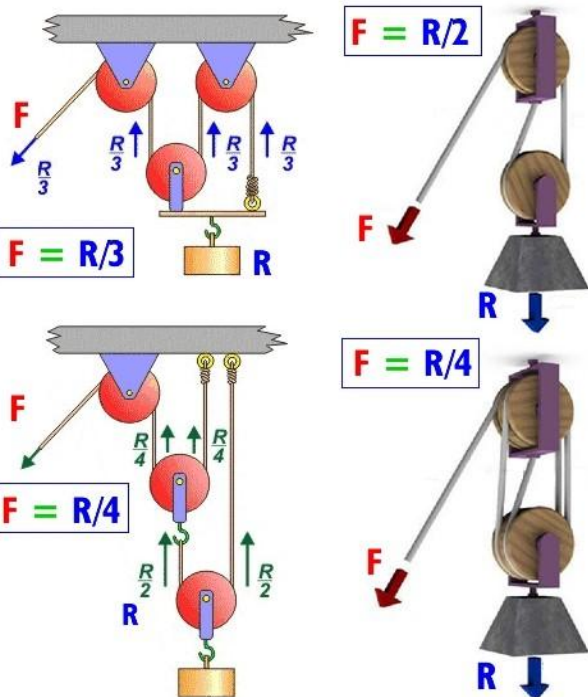
Polipasto

POLIPASTOS O POLEAS COMPUESTAS

Montaje compuesto de varias poleas fijas y móviles. Las poleas fijas se emplean para modificar la dirección de la fuerza que ejercemos sobre la fuerza, mientras que las poleas móviles reducen el esfuerzo a aplicar.

Este tipo de sistema se encuentra en grúas, montacargas, ascensores....

La fuerza necesaria para equilibrar el sistema vendrá dado por el número de poleas, y como estén configuradas. Vemos a continuación algunos ejemplos:



EJEMPLOS DE POLEAS



SISTEMAS DE CORREAS DE TRANSMISIÓN

Sistema polea-correa

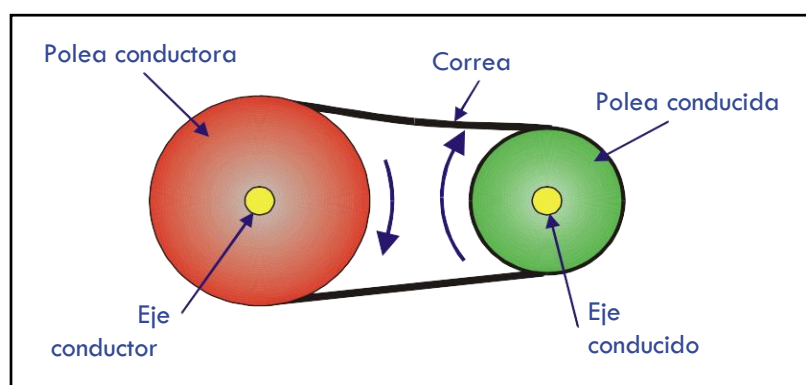
Transmite un movimiento giratorio de un eje a otro, pudiendo modificar sus características de velocidad y sentido. Normalmente los ejes tienen que ser paralelos, pero el sistema también puede emplearse con ejes que se cruzan a 90°

DESCRIPCIÓN

El sistema se compone, básicamente, de dos **ejes** (conductor y conducido), dos **poleas** (conductora y conducida) y una **correa**; a los que se les puede añadir otros operadores como **poleas locas** o **tensores** cuya finalidad es mejorar el comportamiento del sistema.

La finalidad de cada operador es la siguiente:

- El **Eje conductor** es el eje motriz, el que dispone del movimiento que tenemos que transmitir al otro eje.
- El **Eje conducido** es el eje que tenemos que mover.
- **Polea conductora** es la que está unida al eje conductor.
- **Polea conducida** es la que está unida al eje conducido.
- La **Correa** es un aro flexible que abraza ambas poleas y transmite el movimiento de una a otra.



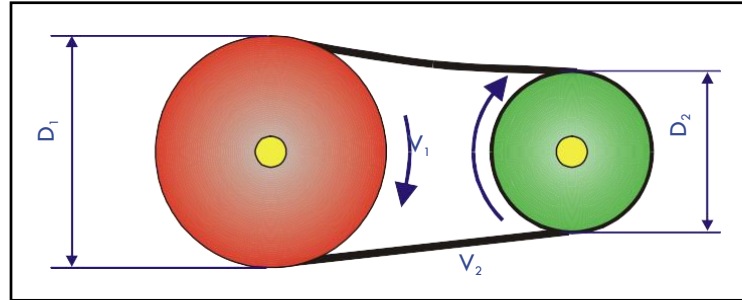
Puede resultar interesante observar que los dos tramos de la correa no se encuentran soportando el mismo esfuerzo de tensión: uno de ellos se encuentra bombeado (flojo) mientras que el otro está totalmente tenso dependiendo del sentido de giro de la polea conductora (en la figura anterior el tramo superior estaría flojo mientras el inferior estaría tenso).

CARACTERÍSTICAS

Este sistema de transmisión de movimientos tiene muchas ventajas: mucha fiabilidad, bajo coste, funcionamiento silencioso, no precisa lubricación, tiene una cierta elasticidad... Por estas razones es

tan usado en aparatos **electrodomésticos** (neveras, lavadoras, lavavajillas...), electrónicos (aparatos de vídeo y audio, disqueteras...) y en algunos mecanismos de los **motores térmicos** (ventilador, distribución, alternador, bomba de agua...).

Su principal desventaja consiste en que cuando la tensión es muy alta la correa puede llegar a salirse de la polea, lo que en algunos casos puede llegar a provocar alguna avería más seria.



RELACIÓN DE VELOCIDADES

La transmisión de movimientos entre los dos ejes está en función de los diámetros de las dos poleas, cumpliéndose en todo momento:

$$D_1 \times N_1 = D_2 \times N_2$$

Definiendo la **relación de velocidades** como:

$$i = \frac{\text{Velocidad eje conductor}}{\text{Velocidad eje conducido}} = \frac{\text{Diámetro polea conducida}}{\text{Diámetro polea conductora}}$$

$$i = \frac{N_1}{N_2} = \frac{D_2}{D_1}$$

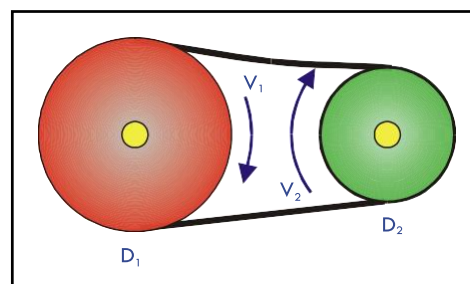
Donde:

D_1 Diámetro Polea conductora

D_2 Diámetro Polea conducida.

N_1 Velocidad de giro Polea conductora

N_2 Velocidad de giro Polea conducida.

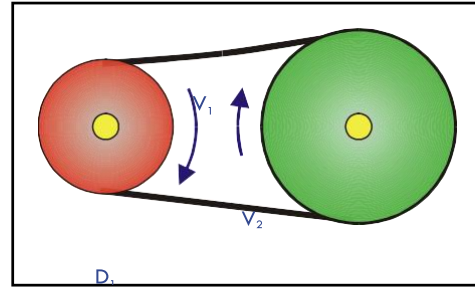


AUMENTO DE LA VELOCIDAD DE GIRO.

Si la *Polea conductora* tiene mayor diámetro que la *conducida*, la velocidad de giro aumenta.

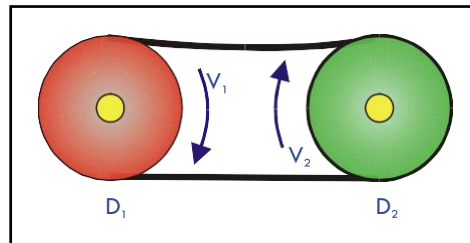
$D_1 = D_2$ $V_1 = V_2$

$D_1 > D_2$ $V_1 < V_2$



DISMINUCIÓN DE LA VELOCIDAD DE GIRO.

Si la *Polea conductora* es menor que la *conducida*, la velocidad de giro del *eje conducido* será mayor que la del *ejeconductor*.



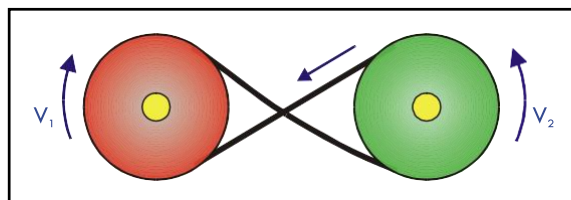
MANTENIMIENTO DE LA VELOCIDAD DE GIRO.

Si ambas poleas tienen igual diámetro, la velocidad de giro de los dos ejes es idéntica.

INVERSIÓN DEL SENTIDO DE GIRO.

Empleando poleas y correas también es posible invertir el sentido de giro de los dos ejes sin más que cruzar las correas.

Con una adecuada *relación de diámetros* se podrá también aumentar ($D_1 > D_2$), disminuir ($D_1 < D_2$) o mantener ($D_1 = D_2$) la velocidad de giro del eje conducido



$D_1 < D_2$ $V_1 > V_2$

MÓDULO 5

COMANDO A DISTANCIA

Mando IR del robot mBot con mBlock

Programación del mando a distancia IR en el robot mBot utilizando mblock.

En este módulo veremos la programación del mando por infrarrojos del robot mBot. Para ello se programará sobre el lenguaje de programación por bloques de mBlock.

Mando a distancia IR

El mando a distancia por infrarrojos es utilizado para controlar el robot mBot de forma remota. El mando está formado por números, letras desde la A hasta la D, flechas de dirección y el botón de configuración. Al funcionar por infrarrojos, a través de un mismo mando se puede controlar diferentes mBot



MAndo IR

Si utilizas el robot mBot en el aula y utilizas mandos a distancia, puedes apuntar directamente sobre el robot en el mando utilizado, y dejar una separación de un par de metros entre robot y robot para que no existan interferencias.

Vamos a programar el mando a distancia sobre el editor el código de nuestro robot. Vamos a utilizar dos variables "izquierda" y "derecha" encargadas de almacenar la velocidad de giro de cada una de las ruedas, de tal forma, que al presionar cualquier flecha este valor irá aumentando o disminuyendo.

El algoritmo funcionará de la siguiente forma; si pulsamos las flechas hacia adelante o hacia atrás, el valor de las variables aumentará o disminuirá respectivamente. Por otro lado, si pulsamos las flechas izquierda o derecha, aumentaremos la variable de giro y disminuirémos la contraria para que haga el giro.

```

cuando mBot(mcore) se pone en marcha
para siempre
si ¿pulsado botón arriba del mando a distancia? entonces
fija Izquierda a 10
fija Derecha a 10
si ¿pulsado botón abajo del mando a distancia? entonces
fija Izquierda a -10
fija Derecha a -10
si ¿pulsado botón izquierda del mando a distancia? entonces
fija Izquierda a 10
fija Derecha a 0
si ¿pulsado botón derecha del mando a distancia? entonces
fija Izquierda a 0
fija Derecha a 10

cuando mBot(mcore) se pone en marcha
para siempre
rueda izquierda a potencia Izquierda %, rueda derecha a potencia Derecha %
    
```

Reto 1: Aumenta la velocidad

En este reto te propongo que modifiques la programación para que el robot aumente la velocidad. Es decir, en vez de cambiar la velocidad de 10 en 10 en sus respectivas condiciones, te propongo que crees una variable llamada "Velocidad" encargada de almacenar el valor de la velocidad, 25 por ejemplo.

