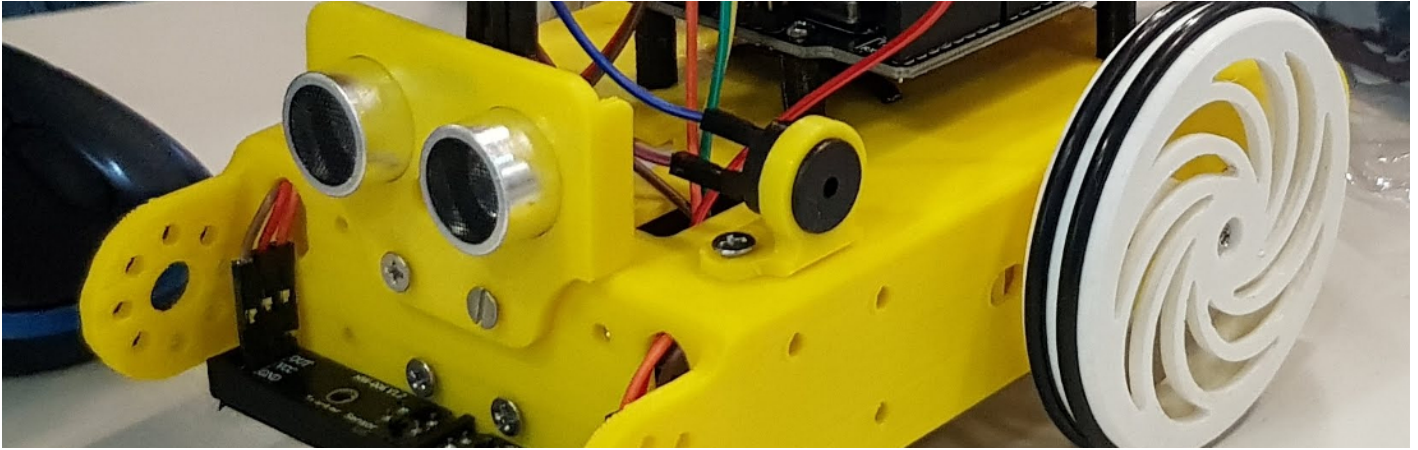


# Regálate Un Mclon

- [Introducción](#)
- [Pensamiento computacional](#)
- [Robótica y accesibilidad](#)
- [1 ¿Cómo es mClon?](#)
  - [1.1 La placa electrónica](#)
  - [1.2 Motores](#)
  - [1.3 Sensores](#)
  - [1.4 Accesorios](#)
  - [1.5 Extras](#)
- [2 Propuesta de Catedu](#)
  - [2.1 Opciones](#)
  - [2.2 Propuesta mClon vs mBot](#)
  - [2.3 Partes impresas](#)
  - [2.4 Lista de material](#)
  - [2.5 18650](#)
- [3 Montaje](#)
  - [3.1 Pasos previos](#)
  - [3.2 Comenzamos](#)
  - [3.3 Cableado esquema](#)
  - [3.4 Cableado motores](#)
  - [3.5 Cableado sensores](#)
  - [3.6 Cableado accesorios](#)

- [3.7 Cableado extras](#)
  
- [4 Testeo](#)
  - [4.1 Con mBlock](#)
  - [4.2 Testeo con mBlock motores](#)
  - [4.3 Testeo con mBlock sensores](#)
  - [4.4 Testeo con mBlock accesorios](#)
  - [4.5 Testeo con mBlock extras](#)
  - [4.5 Testeo completo con mBlock](#)
  - [4.6 Test con IDE](#)
  
- [Muro](#)
- [Grupo telegram mClon](#)
- [Créditos](#)

# Introducción



Fuente: <https://mclon.org> Maria L CC-BY-SA

Tal y como dice <https://tecnologia.org/mclon/que-e-mclon/>

*“ mClon es un robot educativo de bajo costo que imita al mBot y se puede programar con el software mBlock como si fuera un mBot. Está basado en Arduino, por lo que también se puede programar con el IDE de Arduino. Por tanto, es un robot lo suficientemente versátil como para poder adaptarse a **múltiples niveles educativos** , y está especialmente indicado para trabajar en **Educación Secundaria Obligatoria y Bachillerato**. Es un proyecto de hardware y software gratuito”*

Este tutorial NO explica el funcionamiento del Arduino y su programación, sino el montaje de mClon

## OBJETIVOS

1. Conocer la **filosofía Maker**: extensión de la cultura DIY basándose hardware y software libre como elementos de enriquecimiento del conocimiento y adaptación y reutilización de diseños publicados.

2. Conocimientos básicos y habilidad de montaje en **electrónica**: placas, cableados, sensores y actuadores.
3. Conocer las **posibilidades didácticas** de los robots Maker: compatibilidad con los comerciales, programación gráfica y programación en código.

# CONTENIDOS

1. **Conocer mClon**
  1. mClon como sustituto de mBot
  2. La electrónica de mClon
  3. Concienciación de las diferentes opciones de montaje, sus ventajas e inconvenientes.
2. **Propuesta** de montaje de CATEDU: ventajas e inconvenientes
  1. Chasis 3D, la alimentación PowerBank y Placa Robodyn
  2. Extras que se proponen: ventajas didácticas.
3. **Montaje de mClon**
  1. Preparar el material y adecuarlo
  2. Ensamblar las piezas en el chasis
  3. Cableado de los diferentes elementos
4. **Testeo**
  1. Test con programación gráfica mBlock de los diferentes elementos. Conocer las posibles fallas y su solución.
  2. Test completo de todos los elementos con programación gráfica.
  3. Test con programación con código

# Pensamiento computacional

Guía orientativa

[https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0I7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco\\_/embed?start=trueloop=true&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0I7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco_/embed?start=trueloop=true&delayms=3000)

Tenemos un **grupo Telegram Robótica Educativa en Aragón**,  
<https://t.me/roboticaeducativaaragon>



# Robótica y accesibilidad

## 1.- Introducción

Durante mucho tiempo la robótica fue patrimonio de personas y/o instituciones con alta capacidad económica (podían adquirir las placas con microcontroladores comerciales) y capacidad intelectual (podían entender y programar el funcionamiento de las mismas) siempre dentro de los límites establecidos por las marcas comerciales y lo que pudieran “desvelar” de su funcionamiento, vigilando siempre que la competencia no “robara” sus secretos y “copiara” sus soluciones.

Todo esto saltó por los aires en torno a 2005 con la irrupción de un grupo de profesores y estudiantes jóvenes, que decidieron romper con esta dinámica, tratando de poner a disposición de su alumnado microcontroladores económicamente accesibles y que les permitieran conocer su funcionamiento, sus componentes, e incluso replicarlos y mejorarlos. Nació **Arduino** y el concepto de **Hardware Open Source**. Detrás de este concepto se encuentra la **accesibilidad universal**. En un proyecto Open Source todo el mundo puede venir, ayudar y contribuir, minimizando barreras económicas e intelectuales.

Arduino traslada al hardware un concepto ya muy conocido en el ámbito del software, como es el **software open source o software libre**.



### Software libre

Cuando los desarrolladores de software terminan su creación, tienen múltiples posibilidades de ponerlo a disposición de las personas, y lo hacen con condiciones específicas especificadas en una licencia. Esta licencia es un contrato entre el creador o propietario de un software y la persona que finalmente acabará utilizando este software. Como usuarios, es nuestro deber conocer las condiciones y permisos con las que el autor ha licenciado su producto, para conocer bajo qué condiciones podemos instalar y utilizar cada programa.

Existen muchas posibilidades de licencias: software privativo, comercial, freeware, shareware, etc.. Nos centraremos aquí en la de software libre.

GNU (<https://www.gnu.org>) es una organización sin ánimo de lucro que puso una primera definición disponible de lo que es software libre: Software libre significa que los usuarios del software tienen libertad (la cuestión no es el precio). Desarrollaron el sistema operativo GNU para que los usuarios pudiesen tener libertad en sus tareas informáticas. Para GNU, el software libre implica que los usuarios tienen las cuatro libertades esenciales:

1. ejecutar el programa.
2. estudiar y modificar el código fuente del programa.
3. redistribuir copias exactas.
4. distribuir versiones modificadas.

En otras palabras, el software libre es un tipo de software que se distribuye bajo una licencia que **permite a los usuarios utilizarlo, modificarlo y distribuirlo libremente**. Esto significa que los usuarios tienen libertad de ejecutar el software para cualquier propósito, de estudiar cómo funciona el software y de adaptarlo a sus necesidades, de distribuir copias del software a otros usuarios y de mejorar el software y liberar las mejoras al público.

El software libre se basa en el principio de la libertad de uso, y no en el principio de la propiedad. Esto significa que los usuarios tienen la libertad de utilizar el software de la manera que deseen, siempre y cuando no violen las condiciones de la licencia. El software libre es diferente del software propietario, que es el software que se distribuye con restricciones en su uso y modificación. El software propietario suele estar protegido por derechos de autor y solo se puede utilizar bajo los términos y condiciones especificados por el propietario del software.

Recomendamos la visualización de este [video](#) para entender mejor el concepto.

<https://www.youtube.com/embed/nlDVZ816zoI>

Más adelante, entorno a 2015, en Reino Unido, surgiría también la placa **BBC Micro:bit**, con la misma filosofía de popularizar y hacer accesible en este caso al alumnado de ese país la programación y la robótica. También hablaremos de ella.

## 2.- ARDUINO o LA ROBÓTICA ACCESIBLE

Arduino es una **plataforma de hardware y software libre**.

## Hardware libre

Esto significa que tanto la placa Arduino como el entorno de desarrollo integrado (IDE) son de código abierto. Arduino permite a los usuarios utilizar, modificar y distribuir tanto el software como el hardware de manera libre y gratuita, siempre y cuando se respeten las condiciones de las licencias correspondientes.

El hardware libre es un tipo de hardware cuya **documentación y diseño están disponibles de manera gratuita y libre** para su modificación y distribución. Esto permite a los usuarios entender cómo funciona el hardware y adaptarlo a sus necesidades, así como también crear sus propias versiones modificadas del hardware.

Arduino surge como solución al **elevado precio de los microcontroladores** allá por el año 2005. En el ámbito de la educación, los microcontroladores solo se utilizaban en la etapa universitaria, y su coste era tan elevado que muchos proyectos de fin de carrera se quedaban únicamente en prototipos virtuales ya que las universidades no podían proveer a cada estudiante con un microprocesador, contando además que en el propio proceso de experimentación lo más habitual era que una mala conexión hiciera que se rompieran. Otro **gran inconveniente era la dificultad de la programación**. Cada fabricante entregaba su manual de programación, lo que hacía que de unos a otros no hubiera un lenguaje estándar, y la consecuente dificultad de interpretación. Además, su programación era a bajo nivel en lenguaje máquina. Generar una simple PWM requería una ardua y minuciosa secuenciación que podía llevar varias horas hasta conseguir el resultado deseado. Por este motivo, el enfoque de Arduino desde el principio fue ser Open Source tanto en hardware como en software. El desarrollo del hardware fue la parte más sencilla. Orientado a educación, sufre algunas modificaciones frente a los microprocesadores existentes para hacer más fácil su manejo y accesibilidad a cualquier sensor o actuador. El mayor esfuerzo se entregó en todas las líneas de código que hacían posible que ya no hubiera que programar a bajo nivel gracias al IDE de Arduino que incluía bibliotecas y librerías que estandarizaban los procesos y hacían tremendamente sencillo su manejo. Ahora el alumnado para mover un motor, ya no tenía que modificar las tramas de bits del procesador una a una, sino que bastaba con decir que quería moverlo en tal dirección, a tal velocidad, o a equis grados.

Acabábamos de pasar de unos costes muy elevados y una programación muy compleja a tener una **placa accesible, open source y de bajo coste** que además hacía muy **accesible su programación y entendimiento**, características fundamentales para su implantación en educación, hasta tal punto que su uso ya no era exclusivo de universidades, sino que se extiende a la educación secundaria.



Este hecho es fundamental para el desarrollo del Pensamiento Computacional en el aula observándose que su accesibilidad y beneficios son tales, que alcanzan a **centros con alumnado de toda tipología** como la aplicación del pensamiento computacional y robótica en aulas con alumnos de necesidades especiales. Una vez más, aparece el concepto de accesibilidad asociado a esta filosofía Open Source.

A este respecto, recomendamos la lectura de [este interesante blog](#), que tiene por título: "ROBOTIQUEAMOS..." Experiencia de aproximación a la robótica en Educación Especial (CPEE ÁNGEL RIVIÈRE). También recomendamos los trabajos robótica en Educación Especial (CPEE ÁNGEL RIVIÈRE): <http://zaragozacpeeangelriviere.blogspot.com/search/label/ROB%C3%93TICA>



Igualmente, la aparición de Arduino supone una gran facilidad para la aplicación de la robótica y la programación en la atención temprana, donde son numerosas sus aplicaciones desde ayudar a mitigar el déficit de atención en jóvenes autistas, hasta ayudar a socializar a los alumnos con dificultades para ello, o ayudar a alumnos de altas capacidades a desarrollar sus ideas.

Por otro lado su accesibilidad económica lo ha llevado a popularizarse en países de **todo el mundo**, especialmente en aquellos cuyos sistemas educativos no disponen en muchas ocasiones de recursos suficientes, lo que supone en la práctica una **democratización del conocimiento y superación de brecha digital**.

### Filosofía del Arduino ver vídeo

Arduino y su IDE son la primera solución que aparece en educación con todas las ventajas que hemos enumerado, y esto hace que todos los nuevos prototipados y semejantes tengan algo en común, siempre son compatibles con Arduino

Para entender bien la filosofía de Arduino y el hardware libre, os recomendamos este documental de 30 minutos. [Arduino the Documentary](https://player.vimeo.com/video/18390711?h=b5844e7753)

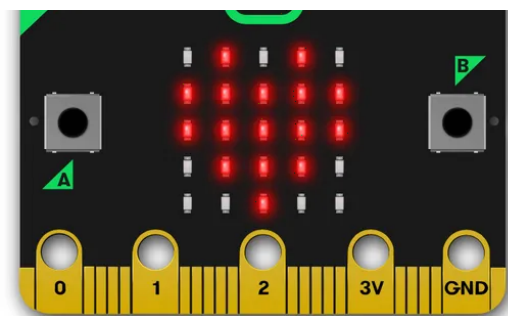
<https://player.vimeo.com/video/18390711?h=b5844e7753>

### Scratch: software libre para el desarrollo del pensamiento computacional

Scratch es un lenguaje de programación visual desarrollado por el grupo Lifelong Kindergarten del MIT Media Lab. Scratch es un software libre. Esto significa que está disponible gratuitamente para todos y que se distribuye bajo una licencia de software libre, la Licencia Pública General de Massachusetts (MIT License). Esta licencia permite a los usuarios utilizar, modificar y distribuir el software de manera libre, siempre y cuando se respeten ciertas condiciones. Entre otras cosas, la licencia de Scratch permite a los usuarios utilizar el software para cualquier propósito, incluyendo fines comerciales. También permite modificar el software y distribuir las modificaciones, siempre y cuando se incluya una copia de la licencia y se indique que el software ha sido modificado. En resumen, Scratch es un software libre que permite a los usuarios utilizar, modificar y distribuir el software de manera libre y gratuita, siempre y cuando se respeten las condiciones de la licencia. De hecho, gracias a que está licenciado de esta forma, han surgido decenas de variaciones de Scratch para todo tipos de propósitos, eso sí, siempre educativos y relacionados con las enseñanzas de programación y robótica

## 3. BBC micro:bit y la Teoría del Cambio

BBC micro:bit, a veces escrito como Microbit o Micro Bit, es un pequeño ordenador del tamaño de media tarjeta de crédito, creado en 2015 por la BBC con el fin de promover el desarrollo de la robótica y el pensamiento computacional entre la población escolar del Reino Unido. Actualmente su uso está extendido entre 25 millones de escolares de 7 a 16 años de más de 60 países.



Tarjeta BBC micro:bit V1. Fuente: <https://microbit.org>. CC BY-

SA 4.0.

Aunque el proyecto fue iniciado por la BBC, su desarrollo fue llevado a cabo por 29 socios tecnológicos de primera línea. Por ejemplo, la implementación del Bluetooth integrado en la tarjeta corrió a cargo de la fundación propietaria de la marca, Bluetooth SIG, una asociación privada sin ánimo de lucro.

**El hardware y el software resultantes son 100% abiertos**, y están gestionados por una fundación sin ánimo de lucro que comenzó a funcionar en el año 2016, la [Micro:bit Educational Foundation](#). La fundación basa sus actuaciones en su Teoría del Cambio,

### Teoría del cambio y más sobre microbit

Teoría del cambio puede resumirse en tres principios:

- El convencimiento de que la capacidad de comprender, participar y trabajar en el mundo digital es de vital importancia para las oportunidades de vida de una persona joven.
- La necesidad de emocionar y atraer a las personas jóvenes por medio de BBC micro:bit, especialmente a las que podrían pensar que la tecnología no es para ellas.
- Diversificar a los estudiantes que eligen las materias STEM a medida que avanzan en la escuela y en sus carreras, para hacer crecer una fuente diversa de talento, impulsando la equidad social y contribuyendo a crear una tecnología mejor.

Para desarrollar sus principios, la fundación trabaja en tres líneas de acción:

- El desarrollo de hardware y software que contribuyan a despertar el entusiasmo en las personas jóvenes hacia la tecnología y hacia las oportunidades que presenta.
- La creación de recursos educativos gratuitos y fáciles de usar que permitan al profesorado enseñar de forma atractiva y creativa.



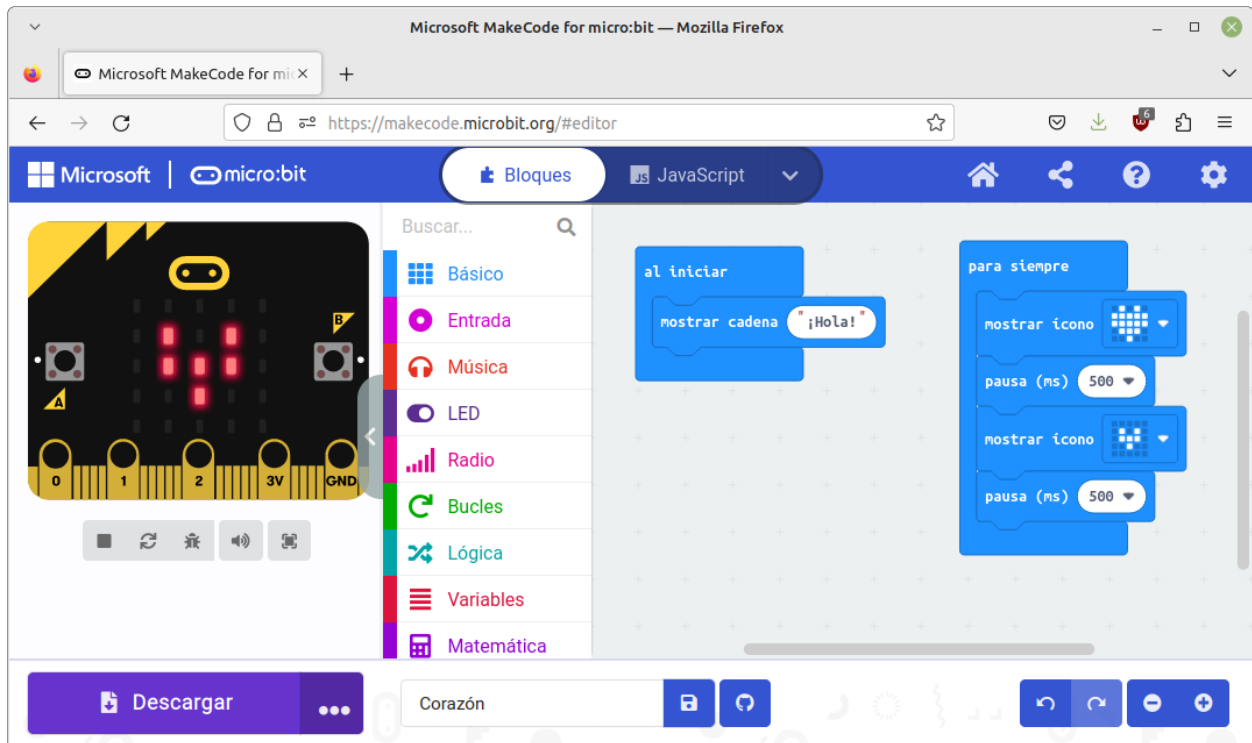
- La colaboración con entidades asociadas que compartan una misma visión para ofrecer programas educativos de alto impacto en todo el mundo.

Uno de los objetivos de la Micro:bit Educational Foundation es llegar a 100 millones de escolares en todo el mundo.

En correspondencia con las líneas de acción y con los principios expuestos, el sistema resultante es muy económico: tanto las placas como los accesorios producidos por terceras empresas tienen un precio muy contenido. Además, dado el carácter abierto del proyecto, están disponibles algunos clones totalmente compatibles, como Elecrow Mbits o bpi:bit. Estos clones son incluso más potentes y económicos que la placa original.

El universo micro:bit destaca por su **alta integración de software y hardware**: basta un clic de ratón para cargar las librerías necesarias para que funcione cualquier complemento robótico, como sensores, pantallas, tarjetas de Internet de las Cosas, robots, casas domóticas, etc.

La programación de la placa se realiza desde un ordenador a través de un navegador cualquiera, estando disponibles **12 lenguajes de programación**. De nuevo, por ser un sistema abierto, existen múltiples soluciones de programación, aunque las más común es [MakeCode](#).



Captura de pantalla del editor MakeCode, <https://makecode.microbit.org/#>.

El sitio web MakeCode permite programar con bloques y también en Python y en Java, traduciendo de un lenguaje a otro instantáneamente. No se necesita ningún registro en la plataforma para poder programar.

Los programas también pueden guardarse descargados en el ordenador compilados en código de máquina. Al subir de nuevo el programa al editor, se realiza una decompilación automática al lenguaje de bloques, Python o Java. Los programas guardados en código de máquina se pueden cargar directamente en micro:bit, que en el escritorio de un ordenador se maneja como una simple unidad de memoria USB.

MakeCode contiene además múltiples recursos como tutoriales, vídeos, fichas de programación, cursos para el profesorado, ejemplos y propuestas de proyectos y experimentos, todo ello en varios idiomas y clasificado por edades desde los 7 años.

Otra solución muy usada para programar micro:bit es [MicroPython](#), creada por Python Software Foundation, otra organización sin ánimo de lucro.

[MicroCode](#) permite que los más pequeños, a partir de los 6 años de edad, programen micro:bit mediante un sistema de fichas dispuestas en líneas de acción. Están disponibles un tutorial introductorio en 20 idiomas, una guía del usuario y muchos ejemplos. El proyecto es de código abierto.



Micro:bit también es programable en **Scratch** con sólo añadir una extensión al editor.

Todos los entornos de desarrollo descritos disponen de un simulador de micro:bit, por lo que ni siquiera resulta necesario disponer de una tarjeta física para aprender a programar.

Una vez realizada la programación, la placa y sus complementos pueden funcionar desconectados del ordenador por medio de un cargador de móvil, una batería externa o un simple par de pilas alcalinas.

### Versiones y características de micro:bit

A pesar de su pequeño tamaño, micro:bit es un sistema potente. Existen dos versiones de la placa. La más moderna, llamada micro:bit V2, tiene las siguientes características:

- Procesador de 64 MHz.
- 512 KB de RAM Flash y 128 KB de RAM.
- Matriz de 5 x 5 LED rojos.
- Dos pulsadores mecánicos y un tercer pulsador de apagado y reset.
- Un pulsador táctil.
- Micrófono y altavoz.
- Acelerómetro y brújula.
- Sensores de luz y de temperatura.
- Comunicación con otras placas por Bluetooth de bajo consumo.
- Alimentación a 3 V o por USB.
- 25 pines de entradas y salidas para conectar motorcitos, sensores, placas de Internet de las Cosas, robots y, en general, cualquier otro tipo de accesorio.
- 200 mA de intensidad de corriente disponibles en las salidas para alimentar accesorios.

## 4.- LA IMPORTANCIA DEL OPEN SOURCE / CÓDIGO ABIERTO EN EDUCACIÓN

La creación, distribución, modificación y redistribución del hardware y software libre así como su utilización, están asociados a una serie de valores que deberían ser explicados en la escuela a nuestros alumnos para dar una alternativa a la versión mercantilista de que cualquier creación es creada para obtener beneficios económicos.



En GNU, pusieron especial énfasis en la difusión del software libre en colegios y universidades, promoviendo una serie de valores fundacionales:

## Valores GNU

### Compartir

El código fuente y los métodos del hardware y software libre son parte del conocimiento humano. Al contrario, el hardware software privativo es conocimiento secreto y restringido. El código abierto no es simplemente un asunto técnico, es un asunto ético, social y político. Es una cuestión de derechos humanos que la personas usuarias deben tener. La libertad y la cooperación son valores esenciales del código abierto. El sistema GNU pone en práctica estos valores y el principio del compartir, pues compartir es bueno y útil para el progreso de la humanidad. Las escuelas deben enseñar el valor de compartir dando ejemplo. El hardware y software libre favorece la educación pues permite compartir conocimientos y herramientas.

### Responsabilidad social

La informática, electrónica, robótica... han pasado a ser una parte esencial de la vida diaria. La tecnología digital está transformando la sociedad muy rápidamente y las escuelas ejercen una influencia decisiva en el futuro de la sociedad. Su misión es preparar al alumnado para que participen en una sociedad digital libre, mediante la enseñanza de habilidades que les permitan tomar el control de sus propias vidas con facilidad. El hardware y el software no debería estar bajo el poder de un desarrollador que toma decisiones unilaterales que nadie más puede cambiar.

### Independencia

Las escuelas tienen la responsabilidad ética de enseñar la fortaleza, no la dependencia de un único producto o de una poderosa empresa en particular. Además, al elegir hardware y software libre, la misma escuela gana independencia de cualquier interés comercial y evita permanecer cautiva de un único proveedor. Las licencias de hardware y software libre no expiran

### Aprendizaje

Con el open source los estudiantes tienen la libertad de examinar cómo funcionan los dispositivos y programas y aprender cómo adaptarlos si fuera necesario. Con el software libre se aprende también la ética del desarrollo de software y la práctica profesional.

### Ahorro



Esta es una ventaja obvia que percibirán inmediatamente muchos administradores de instituciones educativas, pero se trata de un beneficio marginal. El punto principal de este aspecto es que, por estar autorizadas a distribuir copias de los programas a bajo costo o gratuitamente, las escuelas pueden realmente ayudar a las familias que se encuentran en dificultad económica, con lo cual promueven la equidad y la igualdad de oportunidades de aprendizaje entre los estudiantes, y contribuyen de forma decisiva a ser una escuela inclusiva.

## Calidad

Estable, seguro y fácilmente instalable, el software libre ofrece una amplia gama de soluciones para la educación.

### Para saber más

En los años 90, era realmente complicado utilizar un sistema operativo Linux y la mayoría de la cuota del mercado de los ordenadores personales estaba dominada por Windows. Encontrar drivers de Linux para el hardware que tenía tu equipo era casi una quimera dado que las principales compañías de hardware y de software no se molestaban en crear software para este sistema operativo, puesto que alimentaba la independencia de los usuarios con respecto a ellas mismas.

Afortunadamente, y gracias a la creciente presión de su comunidad de usuarios, estas situaciones pertenecen al pasado, y las compañías fabricantes de hardware han tenido que variar el rumbo. Hoy en día tenemos una gran cantidad de argumentos en los que nos podemos basar para dar el salto hacia cualquier sistema operativo basado en Linux. Tal y como podemos leer en [educacionit.com](http://educacionit.com), podemos encontrar las siguientes ventajas:

- Es seguro y respeta la privacidad de los usuarios: Aunque hay compañías linuxeras, como Oracle, Novell, Canonical, Red Hat o SUSE, el grueso de distribuciones y software Linux está mantenido por usuarios y colectivos sin ánimo de lucro. De esta forma, podemos confiar en que una comunidad que tiene detrás millones de usuarios, pueda validar el código fuente de cualquier de estas distribuciones, asegurándonos la calidad de las mismas, compartir posibles problemas de seguridad, y sobre todo, estar bien tranquilos con la privacidad y seguridad de nuestros datos e información personal, aspecto que debería ser crítico y determinante a la hora de trabajar con los datos de menores de edad en las escuelas y colegios.

- Es ético y socialmente responsable: La naturaleza de Linux y su filosofía de código abierto y libre hace posible que cualquier usuario con conocimientos pueda crear su propia distribución basada en otras o probar las decenas de versiones que nos podemos encontrar de una distribución Linux. Este es el caso de Ubuntu por ejemplo. Gracias a esta democratización de los sistemas operativos, incluso han podido aparecer en nuestras vidas nuevos dispositivos basados en software y hardware libre como Arduino y Raspberry Pi.
- Es personalizable: el código abierto permite su estudio, modificación y adaptación a las necesidades de los diferentes usuarios, teniendo así no un único producto sino una multiplicidad de distribuciones que satisfacen las necesidades de los diferentes colectivos a los que se dirijan. Especialmente útiles son las distribuciones educativas libres, que pueden ser adaptadas a las necesidades de las escuelas.
- Está basado en las necesidades de los usuarios y no en las de los creadores de hardware y software
- Es gratis. La mayoría de las distribuciones Linux son gratuitas y de libre descarga
- Es fácil de usar. Una de las barreras que durante años ha evitado a muchos usar Linux es su complejidad. Las distribuciones orientadas al consumo doméstico cumplen los estándares de simplicidad y necesidades que cualquier usuario sin conocimientos de tecnología pueda necesitar. El entorno gráfico es sencillo, intuitivo, e incluso se puede customizar para que se pueda parecer a los más conocidos como Windows y MacOS. Además, vienen con la mayoría de aplicaciones que cualquier usuario puede necesitar: ofimáticas, edición de audio y vídeo y navegación por Internet.
- Es suficiente. Tiene su propio market de aplicaciones. Como el resto de sistemas operativos ya sea para ordenadores o dispositivos móviles, también podemos encontrar un lugar único donde poder descargar cientos de aplicaciones para todos los gustos y necesidades.

Por estas razones, el software libre se ha expandido por toda la comunidad educativa en los últimos años de manera exponencial. Un buen ejemplo de lo que estamos hablando es **Bookstack**, este sistema de edición de contenidos para cursos que utiliza Aularagón así como el uso de **Moodle** como plataforma de enseñanza y aprendizaje. En cuanto a sistema operativo para ordenadores, en Aragón disponemos de nuestra propia distribución Linux: Vitalinux EDU. Tal y como podemos leer desde su página web: **Vitalinux EDU (DGA)** es la distribución Linux elegida por el Gobierno de Aragón para los centros educativos. Está basada en Vitalinux, que se define como un proyecto para llevar el Software Libre a personas y organizaciones facilitando al máximo su instalación, uso y mantenimiento. En concreto Vitalinux EDU (DGA) es una distribución Ubuntu (Lubuntu) personalizada para Educación, "tuneada" por los requisitos y necesidades de los propios usuarios de los centros y adaptada de forma personalizada a cada centro y a la que se ha añadido una aplicación cliente Migasfree. De ésta forma, obtenemos:

1. Un **Sistema Ligero**. Permite "revivir" equipos obsoletos y "volar" en equipos modernos. Esto garantiza la sostenibilidad de un sistema que no consume recursos de hardware innecesariamente ni obliga a la sustitución del hardware cada poco tiempo en esa espiral de obsolescencia programada en la que se ha convertido el mercado tecnológico.
2. **Facilidad en la instalación y el uso** del sistema mediante programas personalizados.
3. Un Sistema que **se adapta al centro** y/o a cada aula o espacio, y no un centro que se adapta a un Sistema Operativo.
4. **Gestión de equipo y del software de manera remota** y desatendida mediante un servidor Migasfree.
5. **Inventario** de todo el hardware y software del equipo de una forma muy cómoda.
6. Soporte y apoyo de una **comunidad** que crea, comparte e innova constantemente.

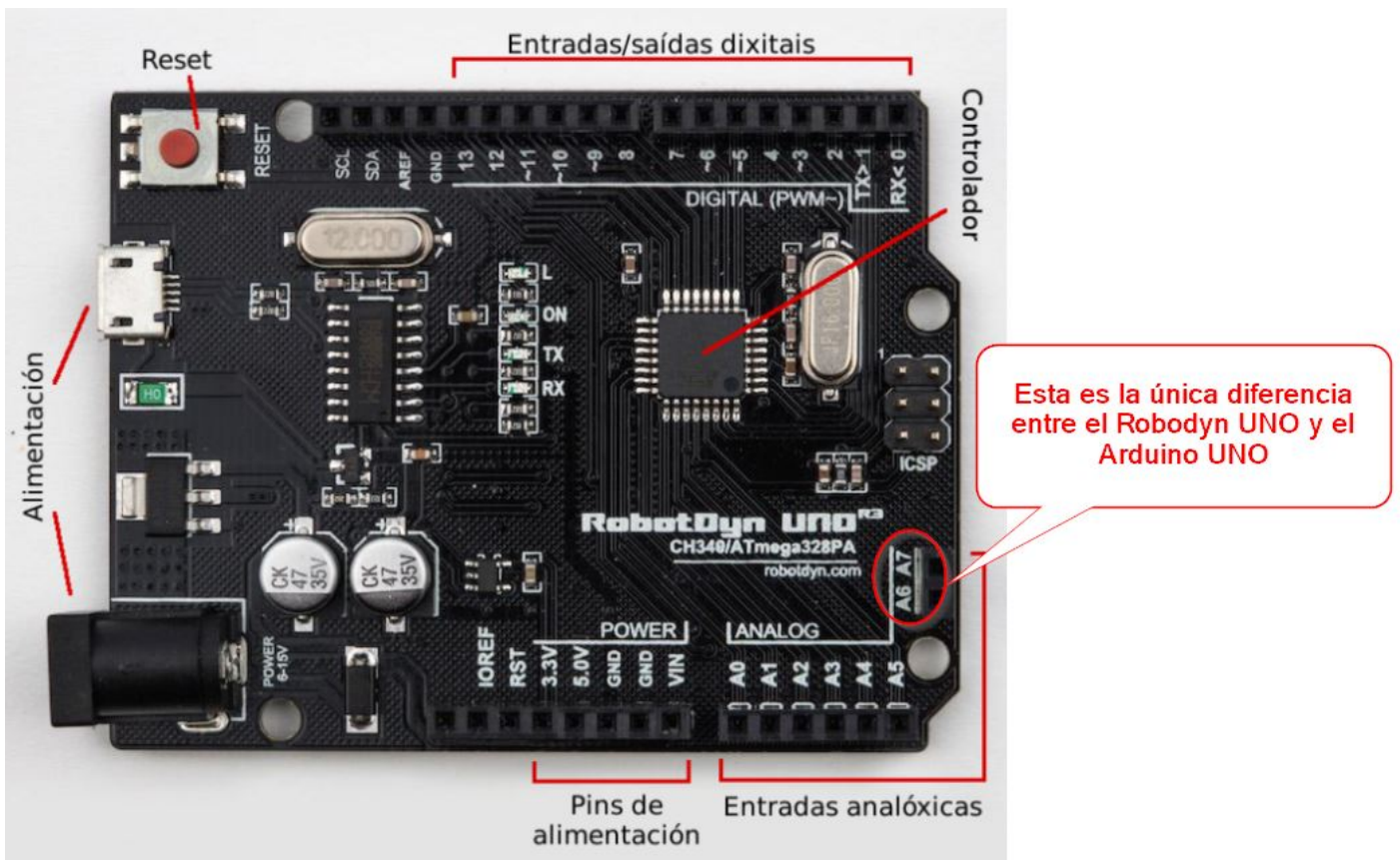
# 1 ¿Cómo es mClon?

1 ¿Cómo es mClon?

# 1.1 La placa electrónica

El robot mClon como mBot son sistemas basados en Arduino. La placa controladora es pues un Arduino, pero como veremos en [opciones](#) en vez de un Arduino UNO vamos a utilizar un **RobotDyn UNO** pues posibilita la utilización del sensor de luz y del botón en placa.

ATENCIÓN: últimamente hay problemas de suministro con esta placa, por eso se ha creado otro tutorial de mClon <https://libros.catedu.es/books/mclon-con-nanoarduino>



Fuente: Adaptado de <https://mclon.org/> Maria L CC-BY-SA



Fuente: <https://mclon.org/> Maria L CC-BY-SA

1 ¿Cómo es mClon?

## 1.2 Motores

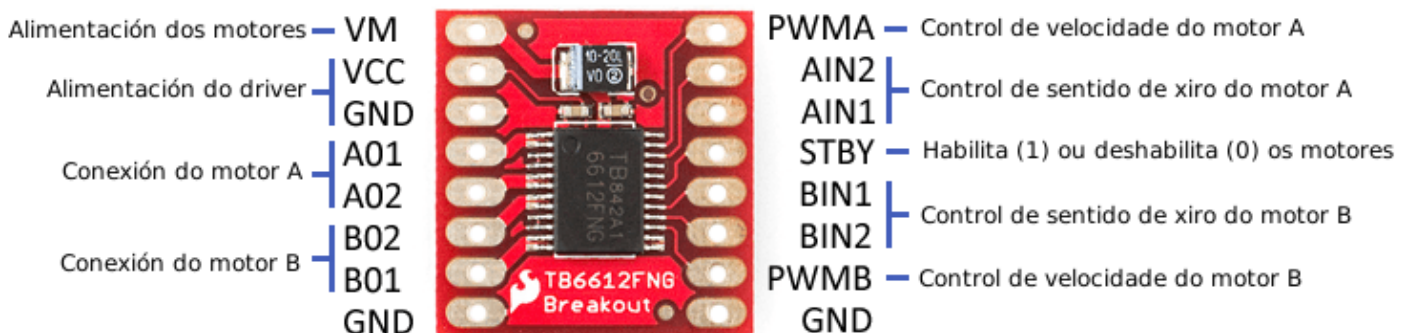
Los motores son de corriente continua sin control de su posición, es decir, sólo podemos controlar su sentido de giro y su potencia.



El control de los motores implicará poner en el escudo Protoboard dos circuitos integrados y su cableado correspondiente, vamos a verlo:

### 1.2.1.- Driver motor B6612FNG

Para realizar el control de los motores, tanto su potencia como su sentido de giro se va a utilizar el controlador **TB6612FNG**



Fuente: <https://mclon.org/> Maria L CC-BY-SA

La potencia de los motores se controla por los pines PWMA y PWMB indicando un valor entre 0 y 255.

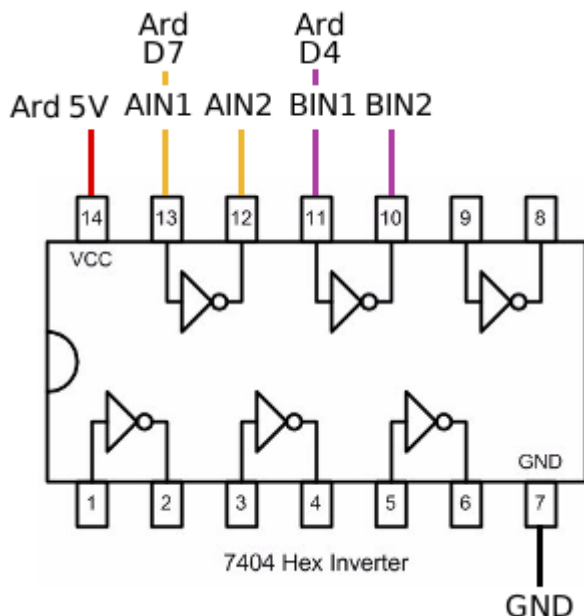
## 1.2.2.- Los giros, una complicación más: 7404

Los giros se controlan con los pines AIN, por ejemplo para el motor A :

GIRO MOTOR	AIN1	AIN2
CLOCKWISE (sentido agujas del reloj)	0	1
ANTICLOCKWISE (contrario)	1	0

Para no gastar dos pines del Arduino para esta función, un truco consiste en gastar sólo uno, y el otro que sea el inverso de ese mismo. Ese truco es lo que utiliza mBot y si queremos compatibilidad mClon y mBot tenemos que hacerlo.

Pero esto implica tener que utilizar un **circuito impreso más**, el 7404 que tiene 4 inversores :



## 1.2.3.- Otra opción para los giros

Tal y como dice la página <https://tecnologia.org/mclon/robotica/o-control-dos-motores/> también se puede utilizar dos transistores y resistencias, (de echo, mBot lo hace así) pero no lo utilizaremos, pues pensamos que es más cómodo el 7404.

1 ¿Cómo es mClon?

## 1.3 Sensores

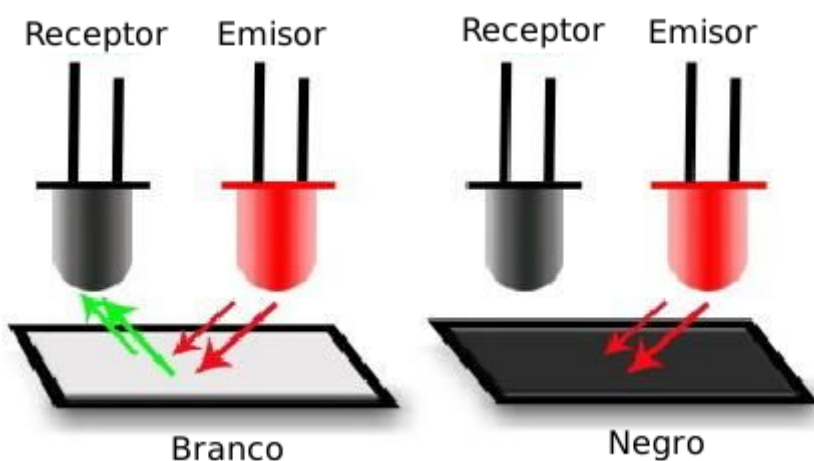
### 1.3.1 Sensor sigue líneas infrarrojos

Utilizamos el tracker sensor TCRT5000



Tal y como hemos explicado en [Partes impresas](#) no tiene que estar cerca de la línea.

En <https://tecnologia.org/mclon/robotica/os-sensores-ir-de-lina/> explica muy bien cómo funciona:



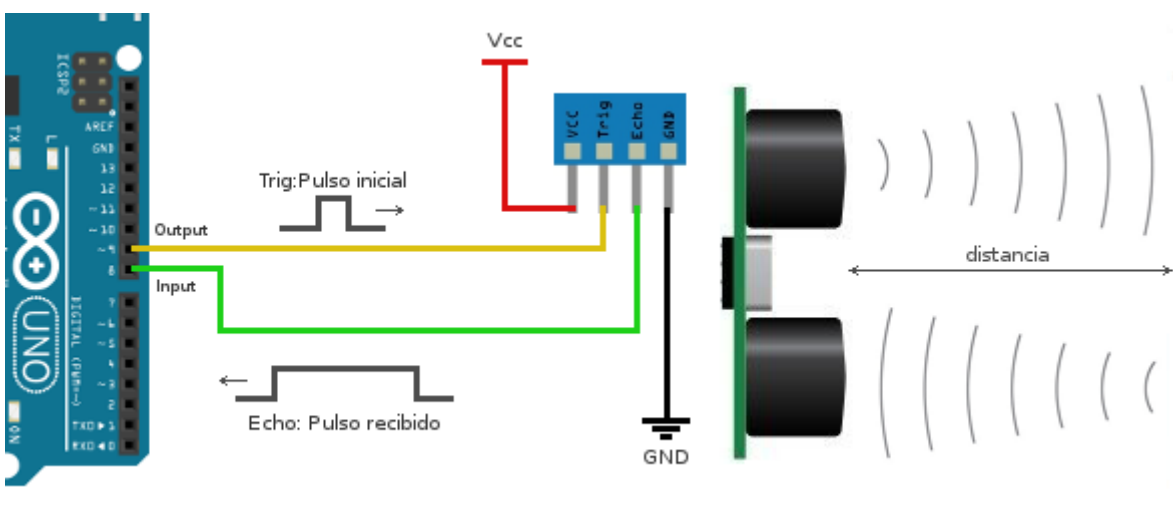
Fuente: <https://mclon.org/> Maria L CC-BY-SA

# 1.3.2 Sensor evita obstáculos ultrasonidos

Se utiliza el sensor HC-SR04



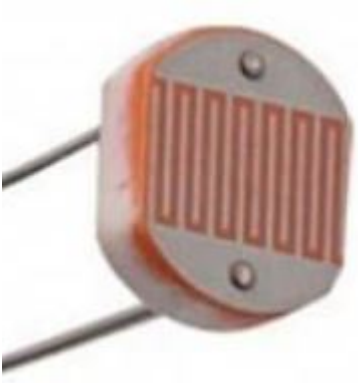
En <https://tecnologia.org/mclon/robotica/o-sensor-ultrasonico/> explica muy bien cómo funciona:





## 1.3.3 Sensor de luz

Es un LDR que conectado en serie con una resistencia nos proporciona un valor de tensión, pues el LDR varía su resistencia con la luz (a más luz, menos resistencia)



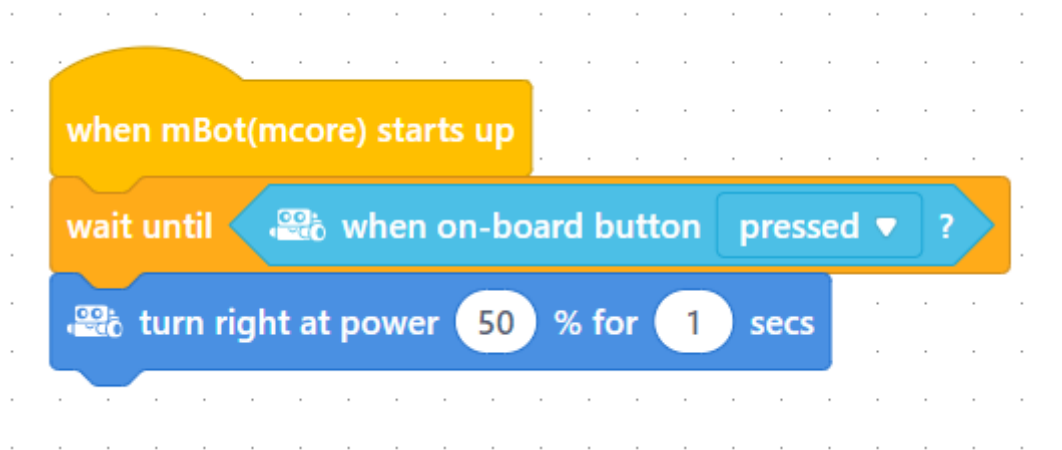
En nuestra propuesta sólo va a tener un LDR, pero se pueden añadir dos LDRs fácilmente, [ver 2.2](#)

1 ¿Cómo es mClon?

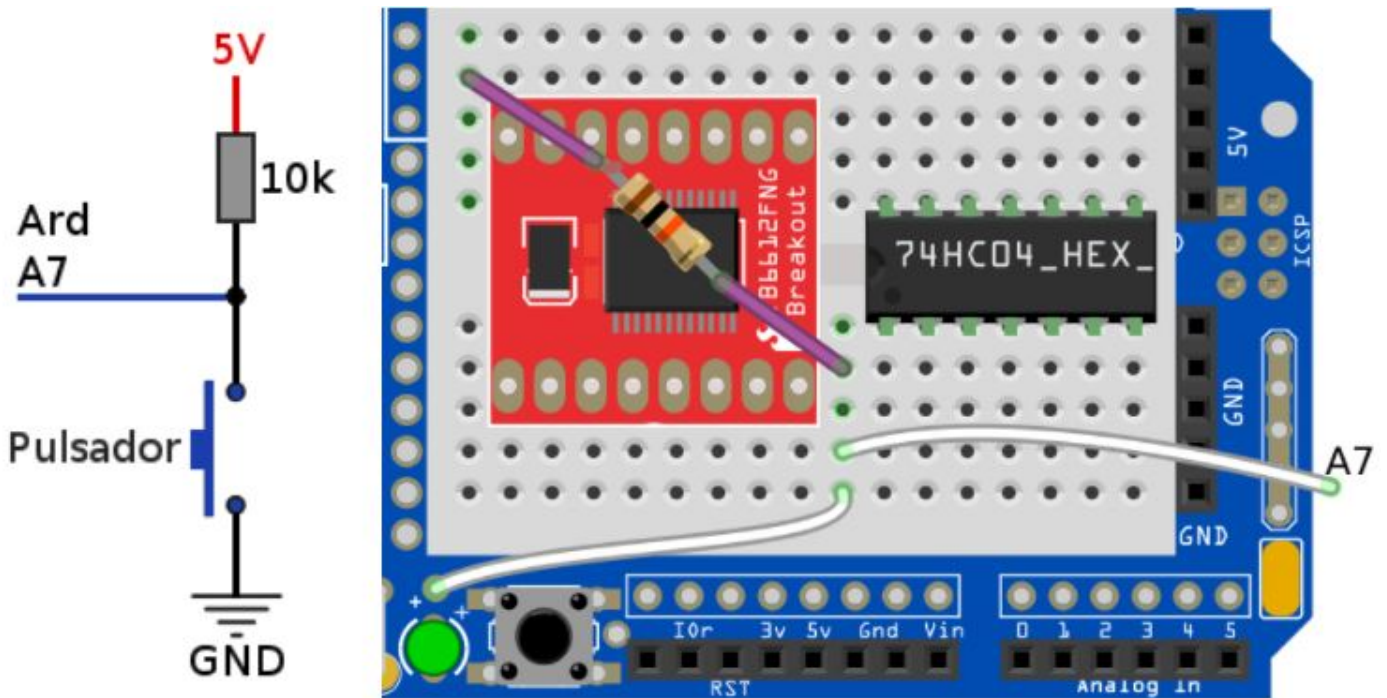
## 1.4 Accesorios

### 1.4.1 Botón en placa

Este botón es muy útil para dar comienzo a los programas, por ejemplo en el siguiente programa hasta que o aprieto el botón el robot no comienza a dar vueltas :



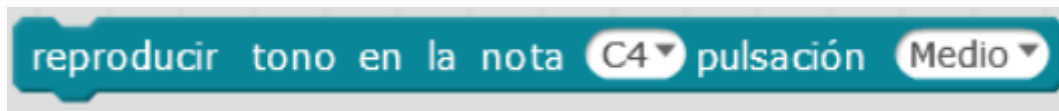
En mClon utilizaremos el botón del escudo Protoboard, pero hay que soldar un cable que lo veremos en **3.1 Pasos previos** para montar el siguiente esquema:



Fuente: <https://mclon.org> Maria L CC-BY-SA

## 1.4.2 zumbador

Sólo se necesita un zumbador **pasivo**, es decir, es un simple altavoz, la instrucción de mBlock ya se encarga de reproducir el tono .



La ventaja de ser pasivo es que podemos reproducir diferentes tonos, para saber más ver [esta página de Luis Llamas](#) . Un zumbador activo reproduce un tono fijado al suministrarle tensión, para saber más [esta página de Luis Llamas](#).

1 ¿Cómo es mClon?

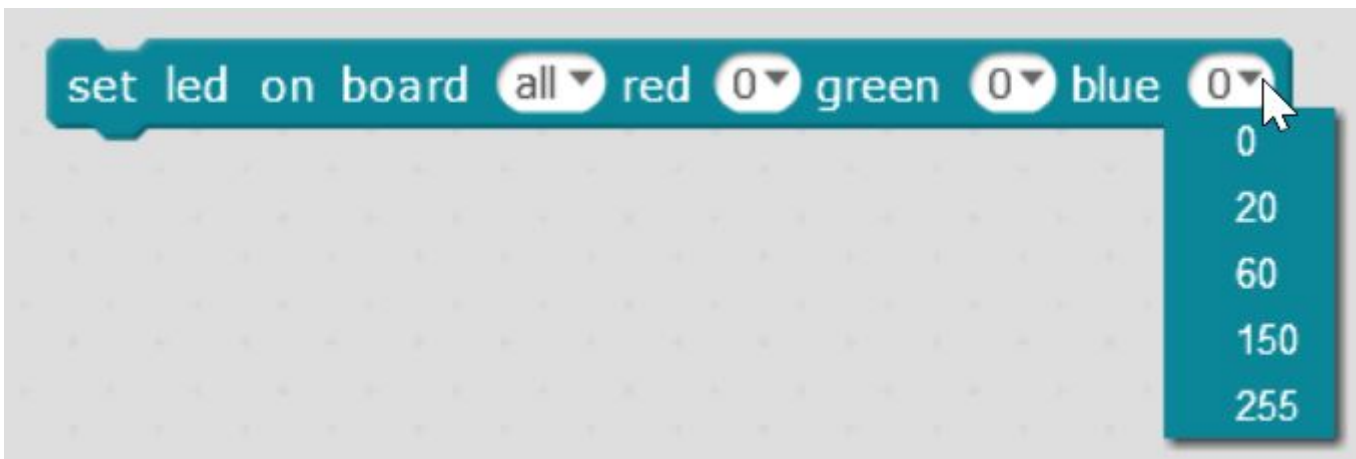
## 1.5 Extras

### 1.5.1 LED RGB

Imitando a mBot, el mClon tiene unos leds RGB que pueden tener diferentes colores.



La instrucción en mBlock permite la combinación de los colores primarios para conseguir la tonalidad que se quiera.



“Xa teño LEDs RGB!! Na páxina <https://t.co/2cUjy6ivcY> podes ver como conectalos e nesta outra <https://t.co/CqI4kUmRbt> explicamos como funcionan e propoñemos algunhas prácticas <pic.twitter.com/vJPyjZ7J6S>

— mClon (@mClonRobot) [November 3, 2018](#)

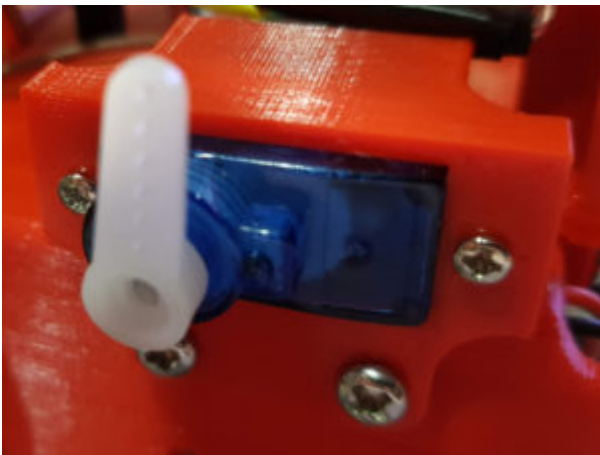
## 1.5.2 Brazo robótico

Los servomotores son motores que sí tenemos control del giro, es decir, que podemos ordenar que gire un ángulo determinado.

Utilizaremos en mBlock la instrucción (como se conectará al D11 de la placa, es el Slot1, si lo conectáramos al D12 sería el Slot2)



Utilizaremos el servo colocado a un lado para insertar el brazo robótico:



Fuente: <https://mclon.org/> Maria L CC-BY-SA

“ Probando servo. Non hai obstáculo que se me resista!

[pic.twitter.com/ZiD7XNDZ3I](https://pic.twitter.com/ZiD7XNDZ3I)

— mClon (@mClonRobot) [November 6, 2018](#)

## 2 Propuesta de Catedu

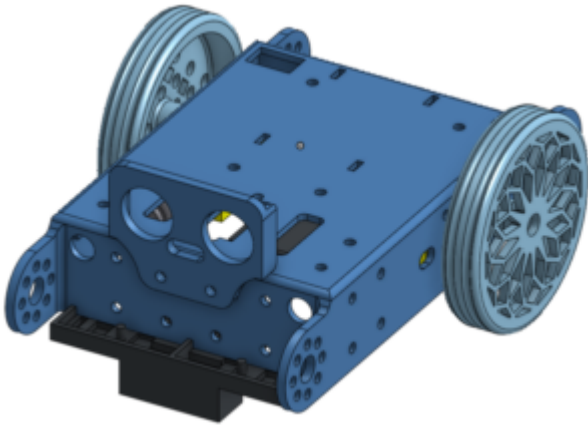
2 Propuesta de Catedu

## 2.1 Opciones

Hay diversas opciones para montar un mClon, todas explicadas en <https://tecnologia.org/mclon/opcions/> la propuesta de CATEDU es la siguiente:

### 2.1.1 Chasis

Elegimos la opción impresa por ser la más adaptable para el docente, en el capítulo de partes impresas lo describimos con detalle.

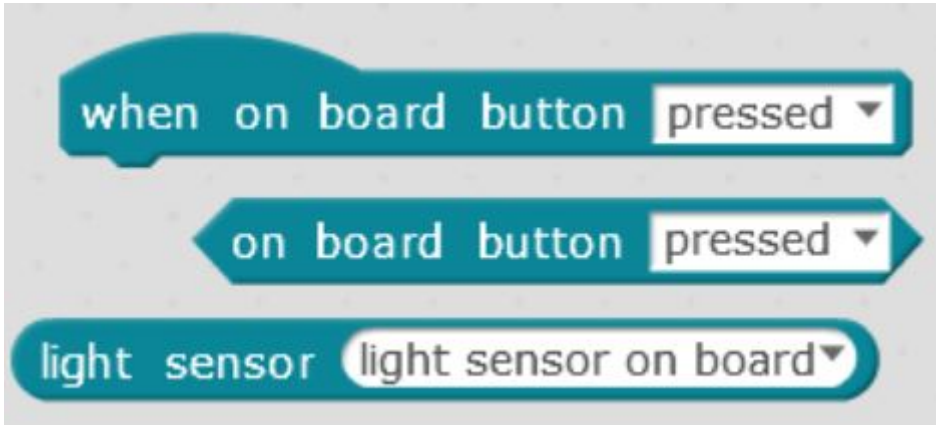


Fuente: <https://mclon.org> Maria L CC-BY-SA

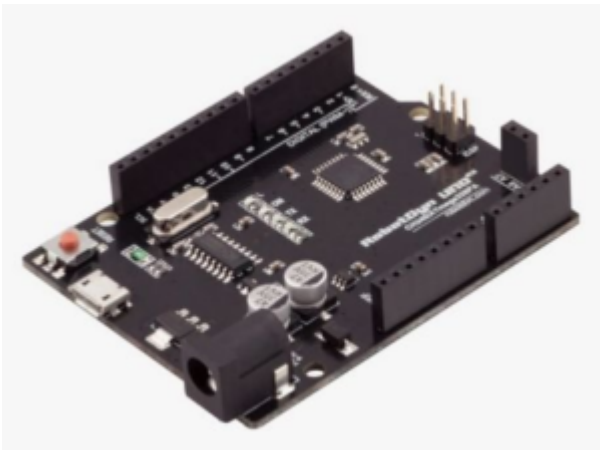
### 2.1.2 Tarjeta electrónica

Estábamos tentados de usar el ARDUINO ONE pues es la que se encuentra más extendida y la más barata **pero tiene una desventaja: no tiene los pines A6 y A7** esto imposibilita las instrucciones:

- sensor luz a bordo
- botón

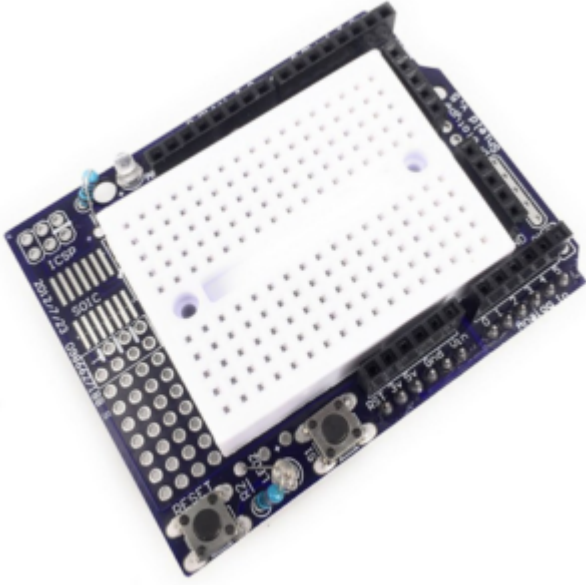


Por lo tanto, preferimos contar con todas las posibilidades de programación de mBot frente a la economía y nos hemos decantado por el **Robotdyn UNO R3**



## 2.1.3 Escudo

Aquí hemos optado por la opción más "Maker" es decir un simple protoboard, esto complica el robot por la **multitud de cableado** no hay que soldar pero si cablear



## 2.1.4 La alimentación

De las tres opciones que ofrece <https://tecnologia.org/mclon/opcion/> lo hemos tenido muy claro: El **powerbank** sencillo, económico, recargable y da mucha potencia.



2 Propuesta de Catedu

## 2.2 Propuesta mClon vs mBot

### Accesorios que vienen con el kit básico de mBot incluido en nuestra propuesta mClon

Hay ciertos accesorios y extras que vamos a añadir a nuestro mClon para ser compatible con el mBot comercial :

- **Sensor ultrasónico** evita obstáculos
- **LED RGB** dos a cada lado del sensor ultrasónico
- **Sensor sigue-líneas**. Dos para cada lado de la línea.
- **Sensor de luz**
- **Zumbador**



*Fuente Makeblock*



# Accesorios que no vienen con el kit básico mBot pero SI que están en nuestra propuesta mClon

## BRAZO

Con mBlock existe la posibilidad de añadir un servo



La propuesta de Catedu es también añadir este extra que le da al mClon unas posibilidades muy creativas :

<https://www.youtube.com/embed/O53jw98uCAo>

---

# Accesorios de mBot que NO están contempladas en el paquete mClon de CATEDU

---

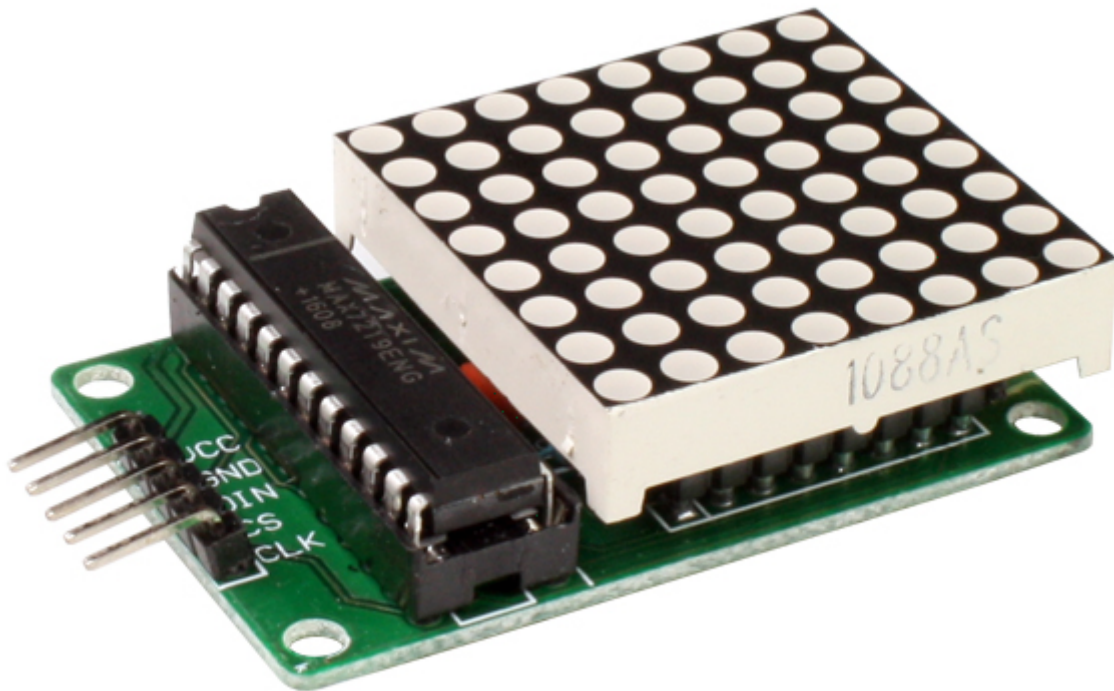
## Matriz LED

mBot tiene una matriz de LEDs 8x16 que permite escribir o dibujar

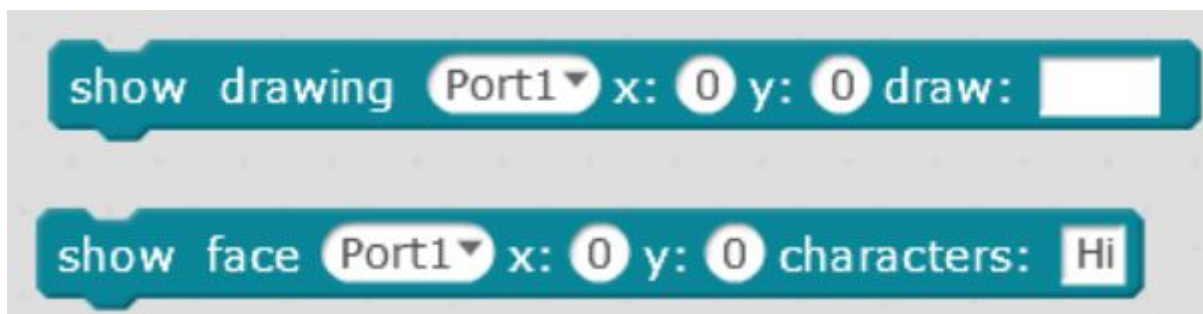


*Fuente: Web de Makeblock.es*

Para mClon existe una matriz similar 8x8



Pero **no es compatible** con la instrucciones de mBlock para la matriz:

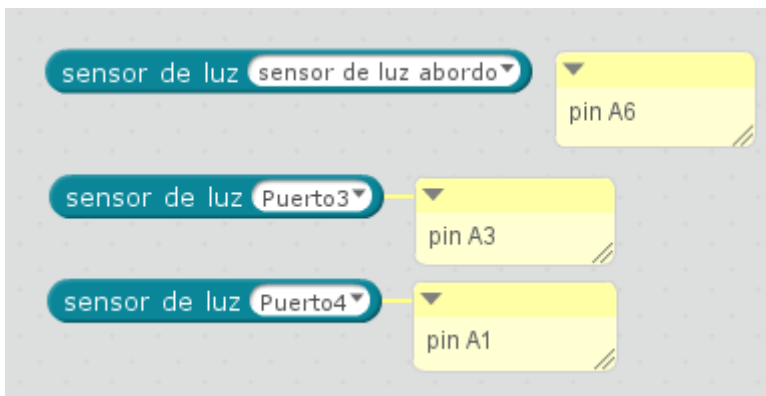


En vez de estas, hay que instalar una librería *MatrixLed* y utilizar las instrucciones correspondientes. **Sólo compatible con la versión mBlock 3 ya en desuso**. Se explica este proceso en <https://tecnologia.org/mclon/64leds/>

Por esta razón **NO** está en la propuesta de mClon de Catedu.

## Dos LDRs

Nuestra propuesta, por simplificar el robot, tiene **un LDR** conectado al A6 que sería el **sensor de luz abordo** pero fácilmente se pueden poner dos LDRs uno al A3 y otro al A1 que serían los equivalentes al **sensor de luz Puerto3** y el **sensor de luz puerto4**



Fuente: <https://mclon.org> Maria L CC-BY-SA

En total se pueden poner **hasta 3 LDRs**.

En la web <https://tecnoloxia.org/mclon/estructura/impresion-3d/> se pueden imprimir unos soportes especiales :



Fuente: <https://mclon.org> Maria L CC-BY-SA

Para más info si lo quieres poner ver <https://tecnoloxia.org/mclon/accesorios/sensor-de-luz/>

Es una propuesta interesante, no lo hemos añadido por simplificar mClon, pero con dos LDRs se pueden hacer cosas interesantes :

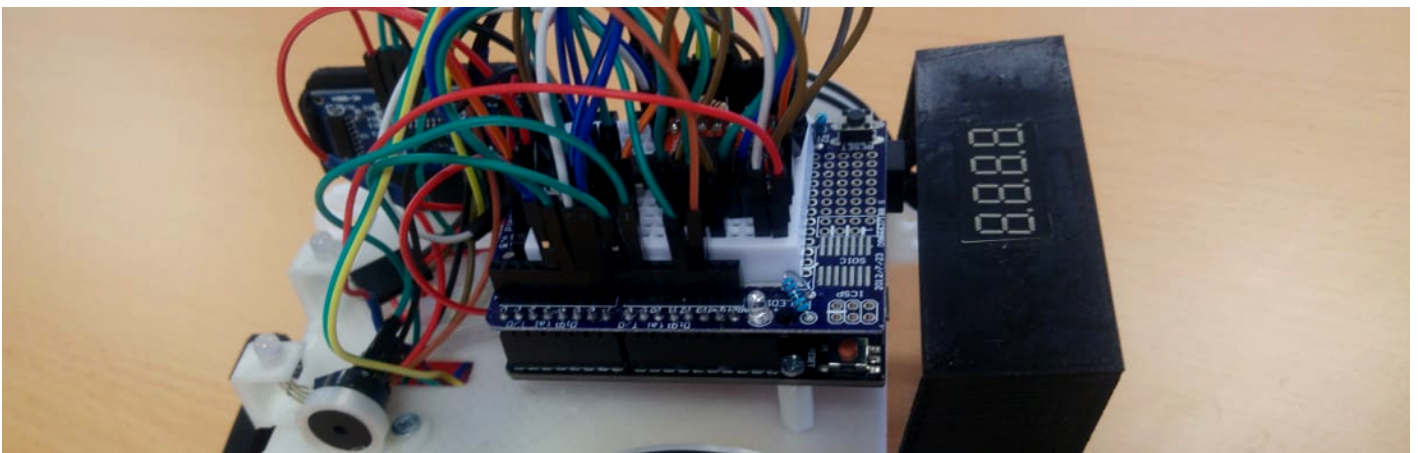
O segue luz non e moi espectacular, pero ten dous ....

LDR [pic.twitter.com/wNyDp1O5Hp](https://pic.twitter.com/wNyDp1O5Hp)

— mClon (@mClonRobot) [November 10, 2018](#)

## Un spoiler con cronómetro

Se trata de una pantalla de cuatro dígitos que puede usarse como cronómetro, contador de eventos, etc..



Fuente: <https://mclon.org> Maria L CC-BY-SA

Para ver cómo se conectaría hay que consultar <https://tecnoloxia.org/mclon/accesorios/aleron-cronometro/>

## Fuera de camino

Esta opción esta documentado también en la Web

<https://tecnoloxia.org/mclon/accesorios/todoterreo/>

“Xa falta pouco para que o modelo todoterreo estea documentado na web. Mentres tanto podes ver como funciona cos dous modelos de oruga, flexible e PLA. [pic.twitter.com/svclPqMJYo](https://pic.twitter.com/svclPqMJYo)

— mClon (@mClonRobot) [April 21, 2020](#)



## Resuelve laberintos

Consiste en cambiar la posición de un siguelíneas para detectar las paredes de enfrente y el sensor ultrasonidos para seguir la pared derecha, en <https://tecnologia.org/mclon/accesorios/resolve-labirintos/> explica muy bien esta opción y las piezas 3d extras para hacerlo:

“Xa resolvo laberintos! Pero, que sexan sinxelos.

Agora documentar e que a xente me replique [pic.twitter.com/QrWEvWiHqs](https://pic.twitter.com/QrWEvWiHqs)

— mClon (@mClonRobot) [November 16, 2018](#)

2 Propuesta de Catedu

## 2.3 Partes impresas

En la web <https://tecnologia.org/mclon/estructura/impresion-3d/> tenemos la última versión de los modelos 3D para descargarlos y los consejos para su impresión. O también en [3D/Imprimibles3D at master · mClon/3D · GitHub](#)

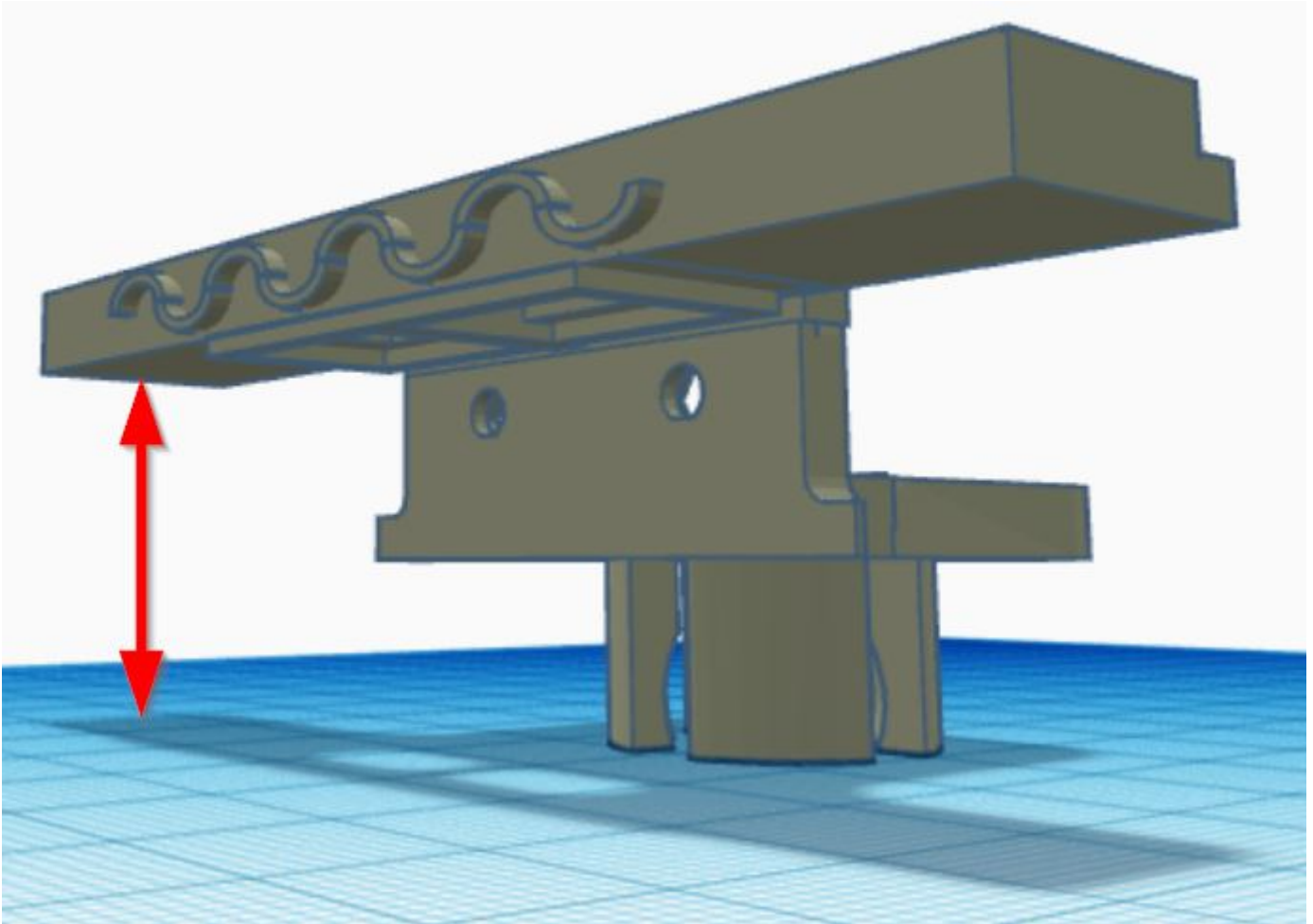
Todos los ficheros disponibles en

[https://drive.google.com/drive/folders/1D8vIUNfCDCvM\\_04RG5HMxtUEGcaBmnTI?usp=sharing](https://drive.google.com/drive/folders/1D8vIUNfCDCvM_04RG5HMxtUEGcaBmnTI?usp=sharing)

- El **Chasis**: Chasis\_mClon\_v2.stl
- El **soporte para ultrasonidos**, que utilizaremos el simétrico, pues hay HC-SR04 que tienen el condensador de cuarzo arriba y otros abajo: ultrasonico\_simetrico.stl
- **Ruedas**, hay con muchos diseños, muy bien conseguidos, nosotros nos hemos decantado por uno sencillo Roda\_5radiosFC.stl
- **Soporte zumbador** ZumbadorSoporte
- **Brazo robótico** con soporte para el servo ServoBrazo.stl
- El **led RGB** el derecho RGB-der.stl y el izquierdo RGB-esq.stl

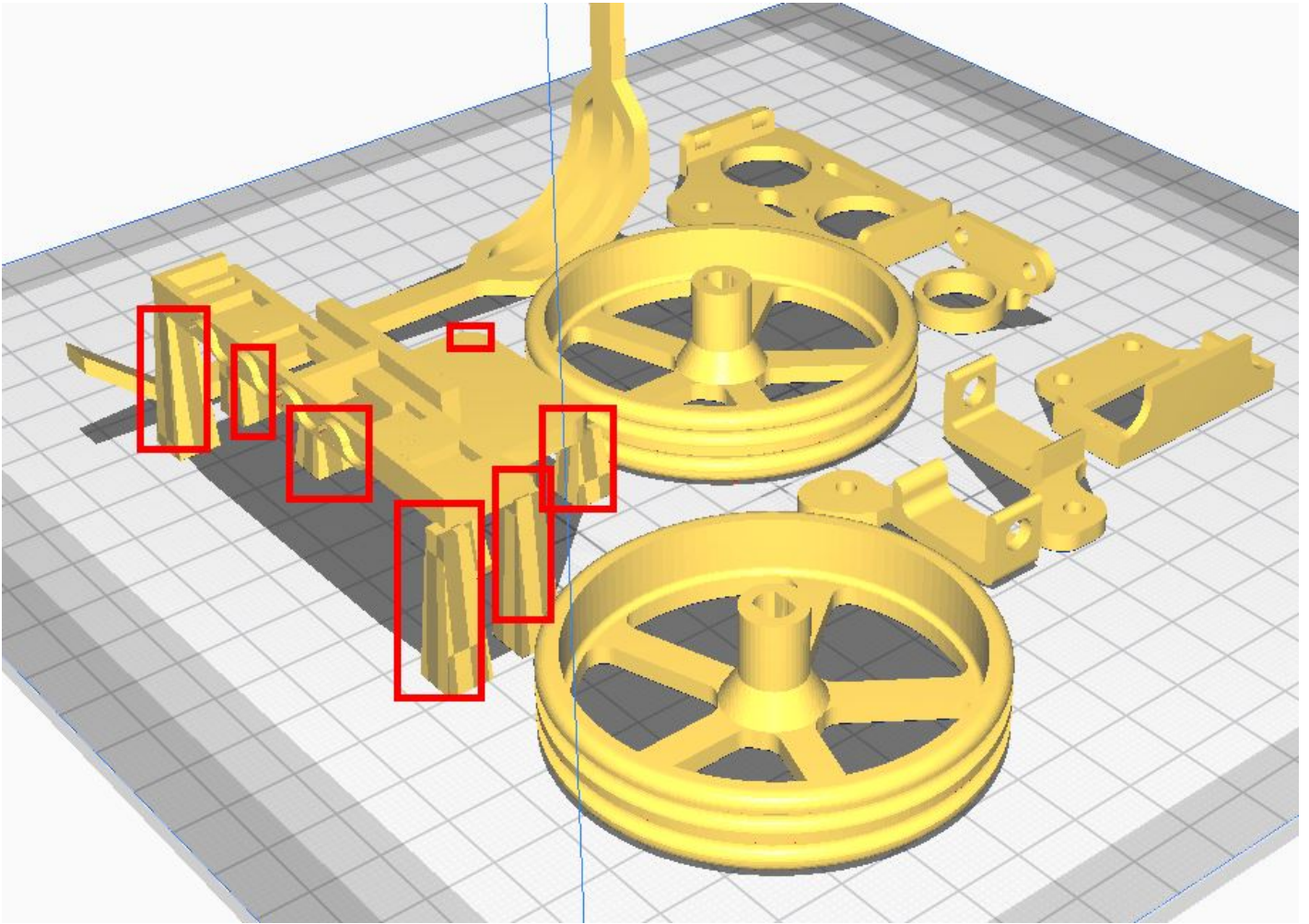
### 2.3.1 ATENCIÓN: EXCEPCIÓN

El **soporte de sensor de línea y bola loca**, no vamos a utilizar el que propone, pues hemos observado (al menos con nuestros sensores de siguelíneas) que **NO** detectaban bien las líneas negras y blancas pues estaban los sensores muy cerca del suelo, por lo que hemos cambiado el diseño original subiéndolo al máximo:



El fichero [SIGUELINEAS-CATEDU.stl](#)

Esta pieza necesita soportes para su correcta impresión :



[todo-menos-chasis.3mf](#)

2 Propuesta de Catedu

## 2.4 Lista de material

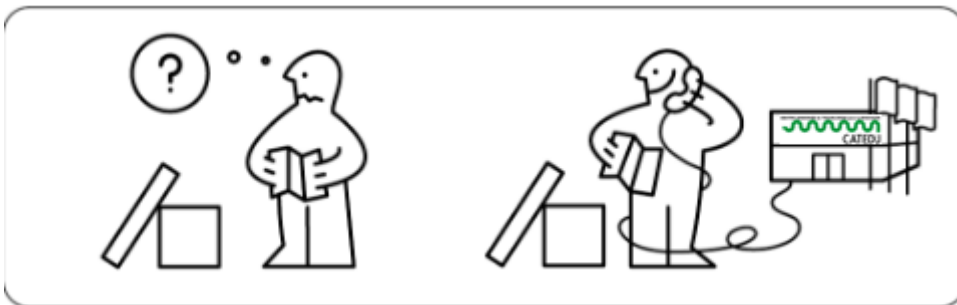
En <https://tecnologia.org/mclon/material/> tienes diferentes opciones y consejos de compra muy útiles.

La propuesta de Catedu se utilizó [esta lista](#) que la iremos actualizando en sucesivas compras.

Las baterías 18650 son **peligrosas**, no son aptas para que el alumno lo manipule, por lo que se entrega el powerbank con la pila montada. Esto necesita una explicación aparte [en esta página](#)

[https://docs.google.com/spreadsheets/d/e/2PACX-1vRqb\\_-nn1kxDSNMYA1xD\\_GHMOVJ5pdI9ufGwCRo8DC\\_xovHAv03YrtBk49CxWIKQ7KAYFfV6gG-WBXVq/pubhtml?gid=0single=true&widget=true&headers=false](https://docs.google.com/spreadsheets/d/e/2PACX-1vRqb_-nn1kxDSNMYA1xD_GHMOVJ5pdI9ufGwCRo8DC_xovHAv03YrtBk49CxWIKQ7KAYFfV6gG-WBXVq/pubhtml?gid=0single=true&widget=true&headers=false)

Si en tu paquete de formación de este robot te falta algo, ya sabes, en [www.catedu.es](http://www.catedu.es) tienes nuestro correo electrónico o nuestro WhatsApp (preferible):



2 Propuesta de Catedu

## 2.5 18650

Son las baterías que podemos encontrar en los portátiles, coches eléctricos... son de Litio, por lo tanto **no** tienen el efecto memoria de las Ni-Mh de las pilas recargables y tienen una alta capacidad.

¿Por qué no se utilizan en vez de las pilas recargables si son mejores?

Porque son **PELIGROSAS** :

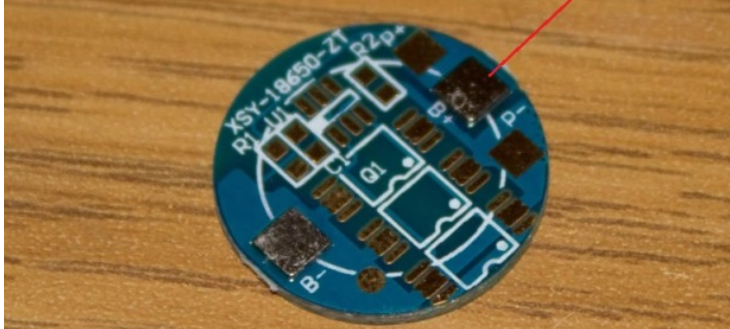
## EXPLOSIÓN

En este vídeo puedes ver la explosión que se genera por un simple cortocircuito (colocación de un pequeño metal encima), imagínate esta explosión en un recinto cerrado, en una linterna, en un armario... o peor: en clase con el alumnado.

Here is a UltraFire 18650 4000mAh Li-ion battery that I bought off eBay



This is the PCB board on the bottom of the cell.  
Probably a protection board, without any protection...



So this is what is under the green lable, a cell..  
Here I have created a controlled short circuit, a common problem...



El vídeo es más impresionante : <https://www.youtube.com/watch?v=ZTzEHsJVZhA>

<https://www.youtube.com/embed/ZTzEHsJVZhA>

## La 18650 del mClon viene montado dentro de la caja PowerBank

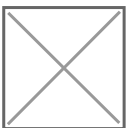
¿Por qué? Por estos peligros, para evitar su manipulación. Si abres la caja powerbank y conectas la batería al revés o dejas que lo manipulen los alumnos es tu responsabilidad.

Si alguna vez quieres reciclar las baterías del portátil, tienes que evitar estos cortocircuitos, su explosión puede causar daños graves [ver](#)

Estas baterías no son aptas para que el alumnado los manipule luego en Alfabot **ES IMPORTANTE QUE LAS BATERÍAS SÓLO LO MANIPULE EL DOCENTE.**

## CUIDADO CON NO INTERCAMBIAR LA POLARIDAD

**OJO ESTAS PILAS SON PELIGROSAS SI SE CORTOCIRCUITAN O NO SE RESPETA LA POLARIDAD, PUEDEN LLEGAR INCLUSO A EXPLOTAR.** Y para complicarlo, no se ve bien (los símbolos + y - de las 18650 soy muy pequeños) y en Alfabot hay una contradicción, los símbolos de fuera en la placa no coinciden con los símbolos de dentro grabados en el portapilas ¿cuales son los verdaderos?: Los de fuera. Para que quede claro aquí tienes un dibujo:



## CUIDADO CON LAS COMPRAS

Se pueden encontrar en tiendas online con cargador incluido. pero hay que tener precauciones.No te fies de las muy baratas, pues hay algunas que [son falsas, LES PONEN HARINA..](#)



# BATERIAS PROTEGIDAS Y NO PROTEGIDAS

¿Qué es eso de la protección? La protección **no quiere decir que estas protegido frente a una explosión**, sino que están protegidas frente a que se descarguen del todo o esten mucho tiempo cargandose, alargando la vida de la batería. Añaden un chip entre la batería y el exterior que desconecta la batería cuando se alcanza valores críticos tanto por abajo cerca del 0% de carga como por arriba cerca del 100%.

Algunas están protegidas, pero lo normal es que no. [Aquí para ver si la pila es protegida o no.](#)

**Las del AlphaBot NO ESTAN PROTEGIDAS** ¿Por qué? porque las protegidas miden 67mm y no caben tiene que ser de 65 mm.

**Las del mClon NO ESTAN PROTEGIDAS** ¿Por qué? porque no caben en la caja powerbank, tiene que ser de 65 mm

# 3 Montaje

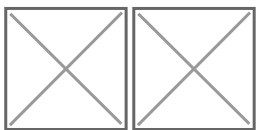
3 Montaje

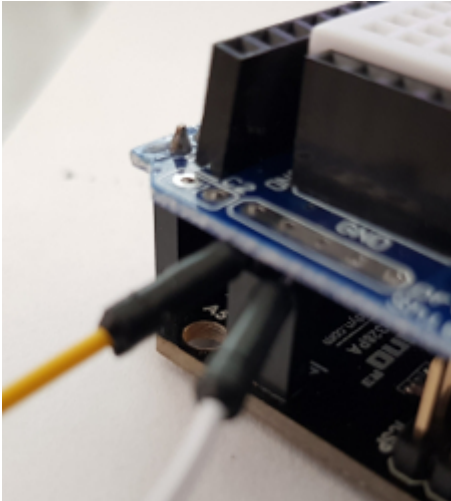
## 3.1 Pasos previos

Los pasos al detalle lo explica en <https://tecnologia.org/mclon/estructura/pasos-previos/> pero con la propuesta de Catedu sólo tenemos que seguir los siguientes pasos:

### 3.1.1 Lo que tienes que hacer tú

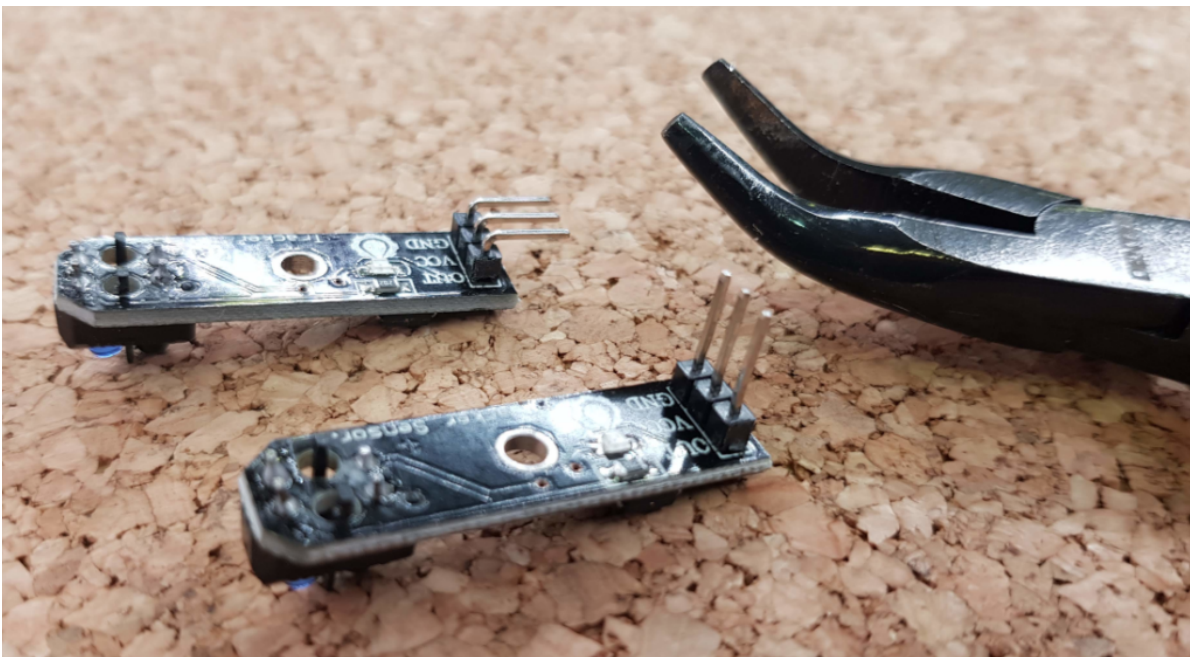
Al insertar el escudo, no nos queda visibles los pines A6 y A7 del Robodyn por lo tanto tenemos





Fuente: <https://mclon.org> Maria L CC-BY-SA

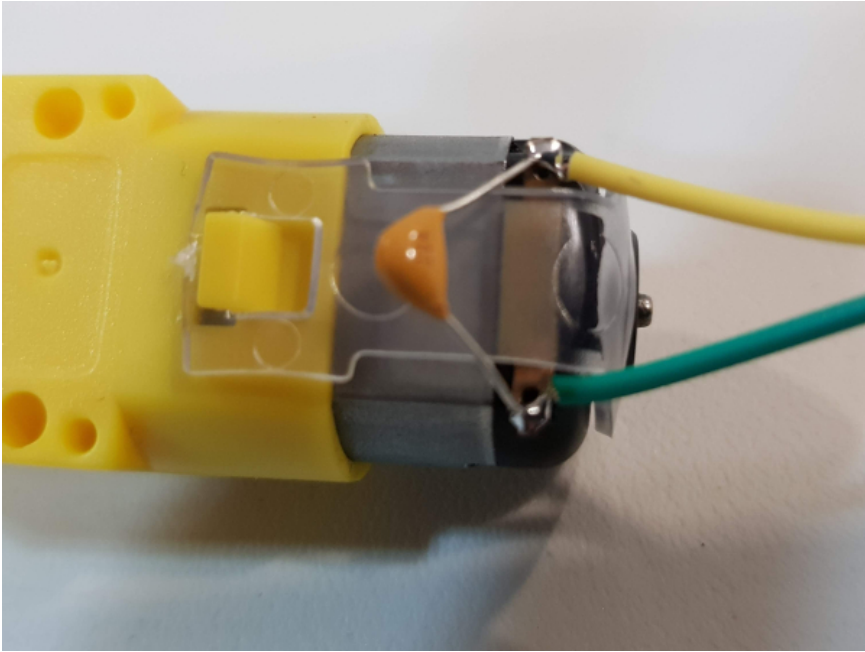
También tienes que doblar los pines de los sensores siguelíneas para que queden perpendiculares al sensor:



Fuente: <https://mclon.org> Maria L CC-BY-SA

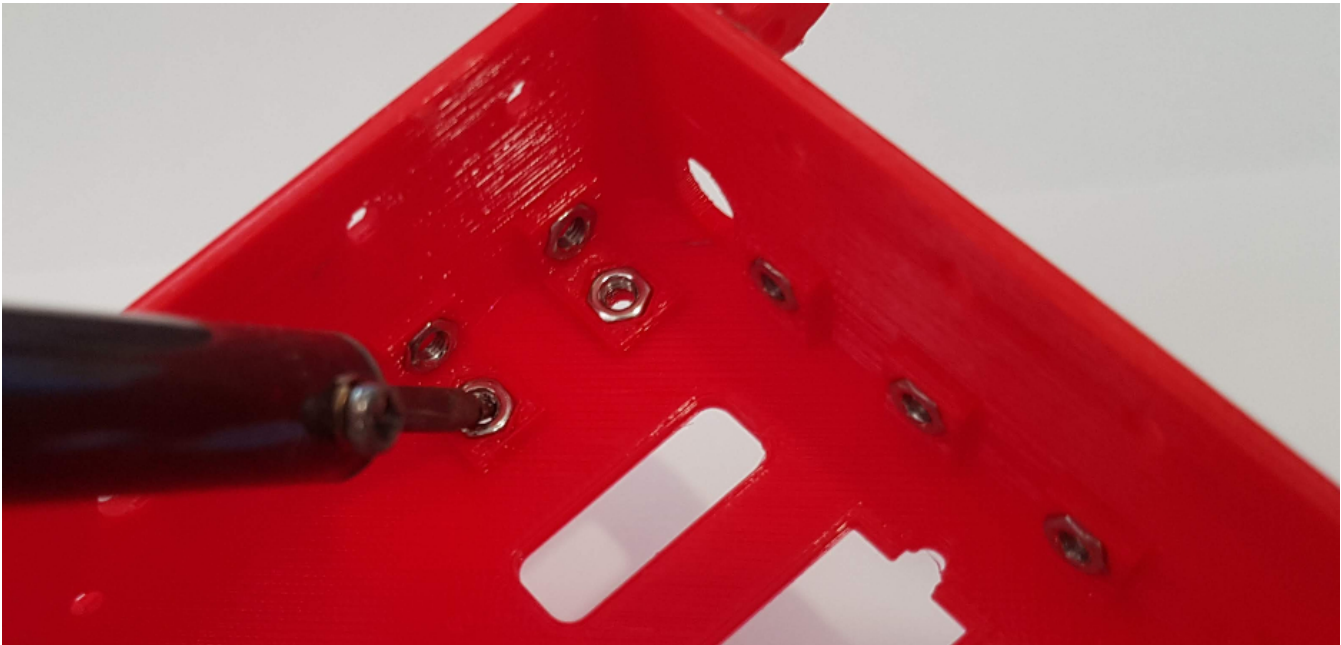
## 3.1.2 Recomendaciones

Para evitar que los picos de los motores afecten a la electrónica de la placa, es recomendable soldar un condensador de  $0.1\mu\text{F}$  en los motores :



Fuente: <https://mclon.org> Maria L CC-BY-SA

También es conveniente que con un soldador caliente fijes los tornillos en las piezas impresas 3D, te facilitará el montaje (no te pases calentando) o si el orificio es muy grande, usar un pegamento para fijar la tuerca a la pieza 3D:

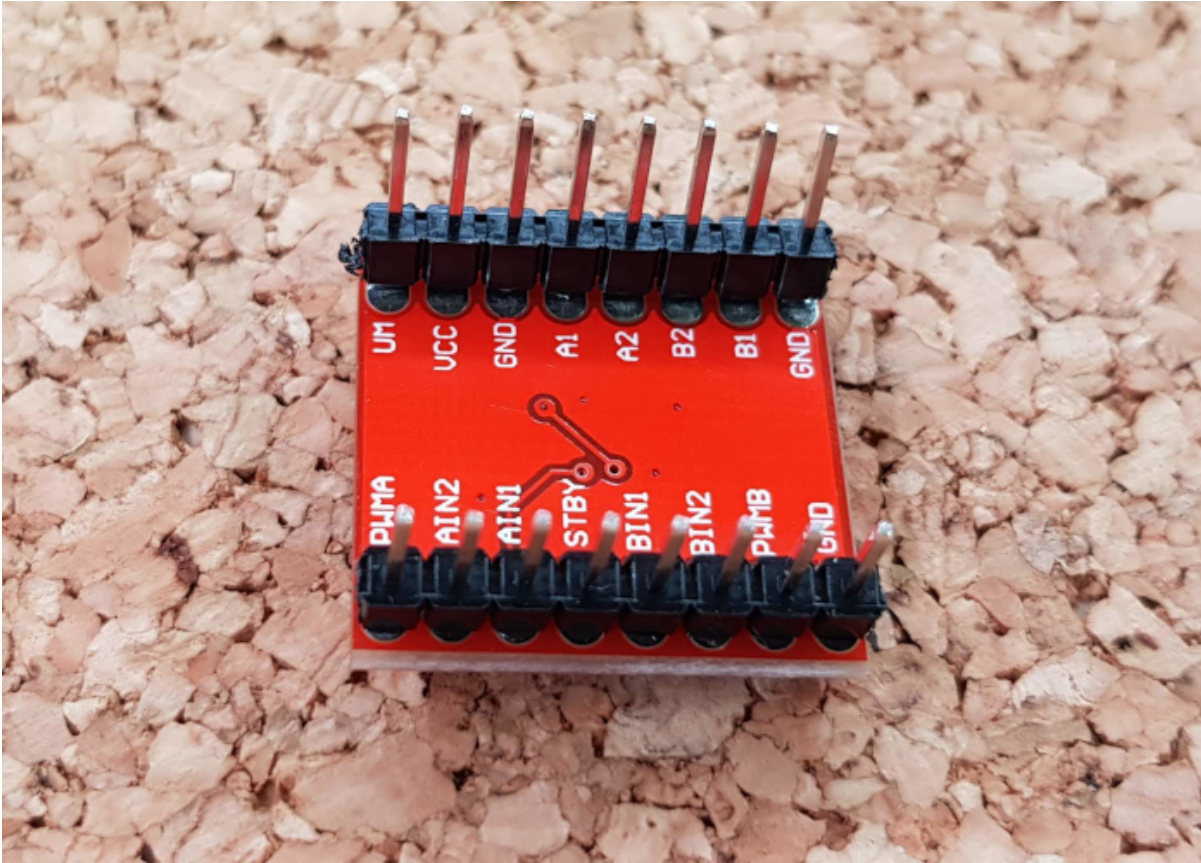


Fuente: <https://mclon.org> Maria L CC-BY-SA

### 3.1.3 Ya realizado por Catedu

Soldado e



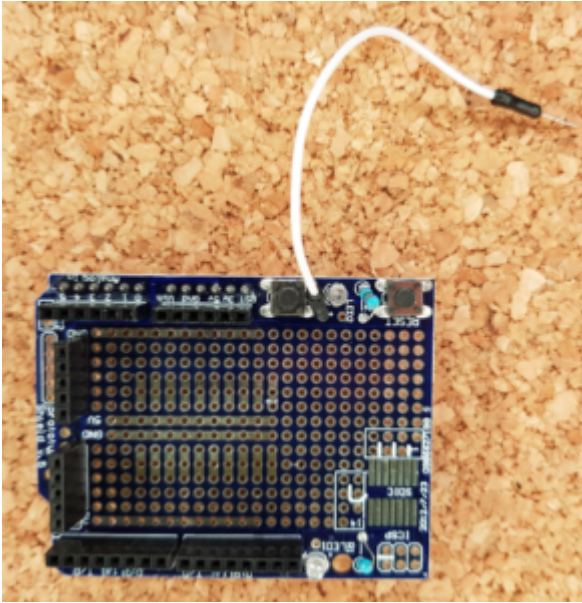


Fuente: <https://mclon.org> Maria L CC-BY-SA

La soldadura

Soldar aquí





Fuente: <https://mclon.org> Maria L CC-BY-SA

La soldadura entre Echo y Trg del sensor US



Fuente: <https://mclon.org> Maria L CC-BY-SA

3 Montaje

## 3.2 Comenzamos

El montaje de este robot es complejo, ánimo !

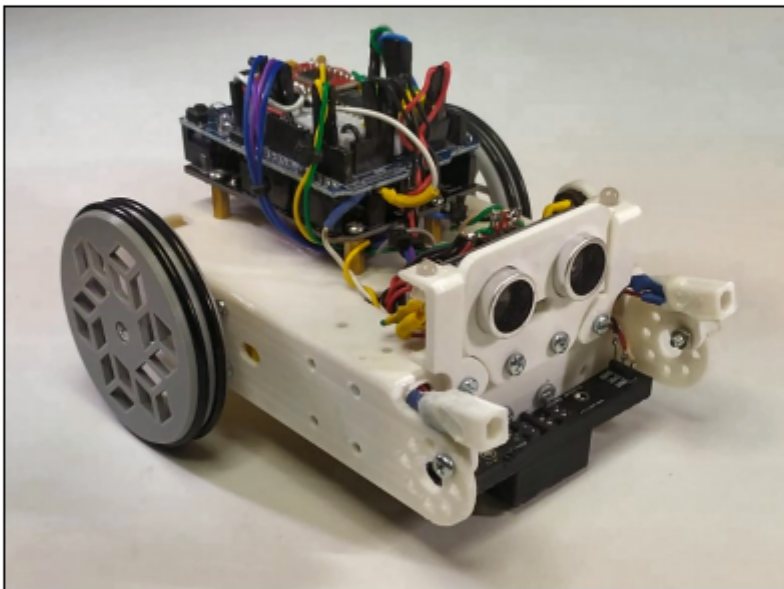


### 3.2.0 Recomendaciones

Te dejamos el enlace al libro [GUIA DE MONTAJE DEL ROBOT MCLON CC-BY-SA](#) de Tino Fernandez Cueto

# GUÍA DE MONTAJE DEL ROBOT MCLON

Modelo: "El Calvo"

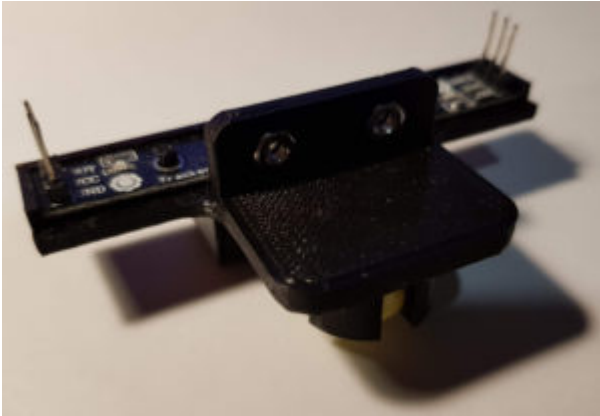


*"La tecnología siempre debe de ir acompañada de una metodología"*



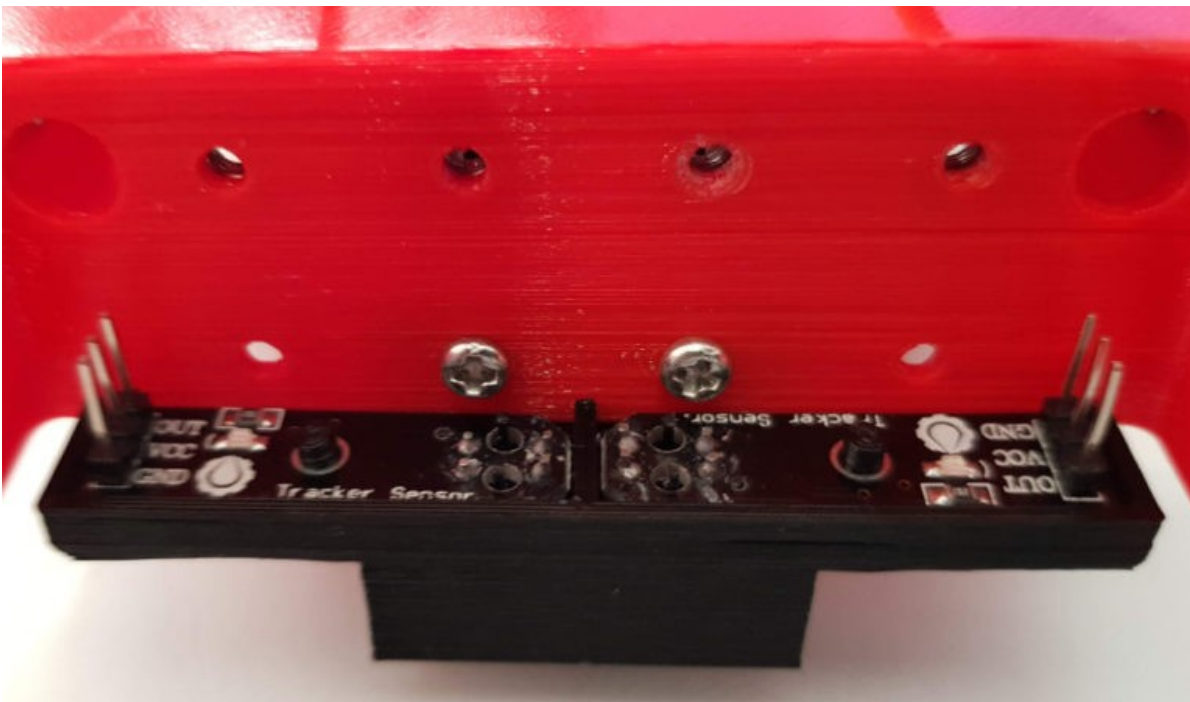
## 3.2.1 Seguidor siguelineas y bola loca

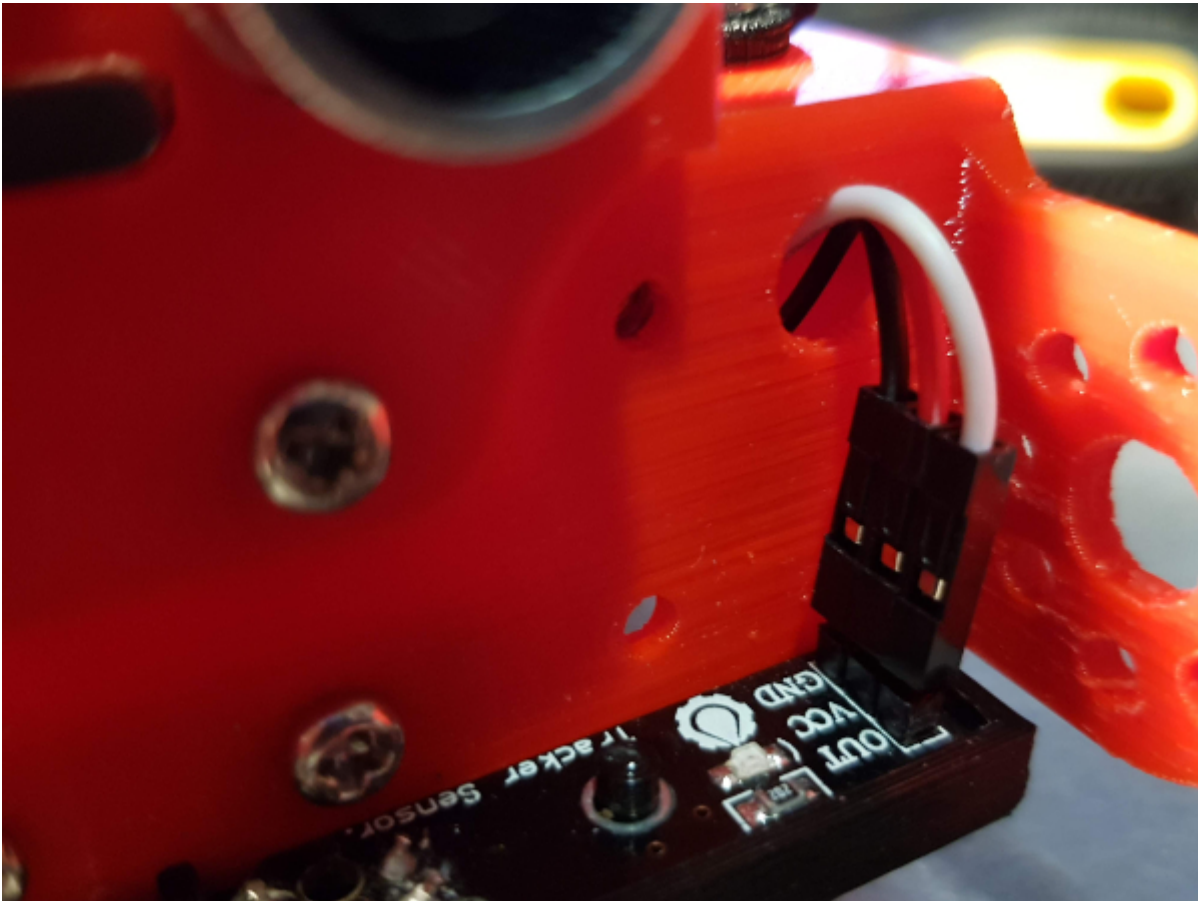
Ponemos la bola loca y los seguidores en el soporte con los leds hacia abajo:



Fuente: <https://mclon.org> Maria L CC-BY-SA

Atornilla el soporte al chasis y conecta los cables pasándolos por el orificio:

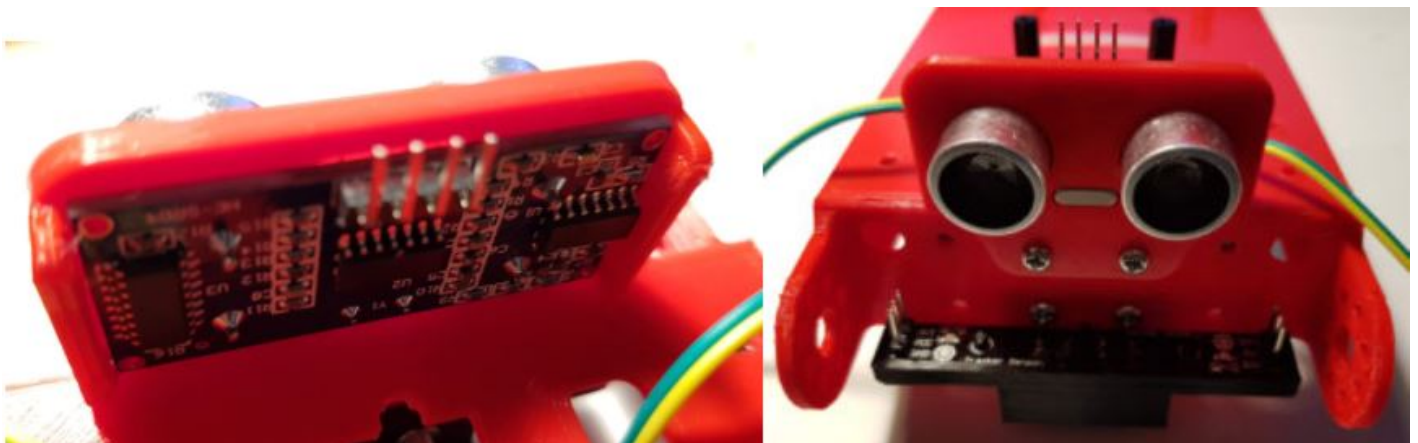




Fuente: <https://mclon.org> Maria L CC-BY-SA

## 3.2.2 Sensor ultrasonidos

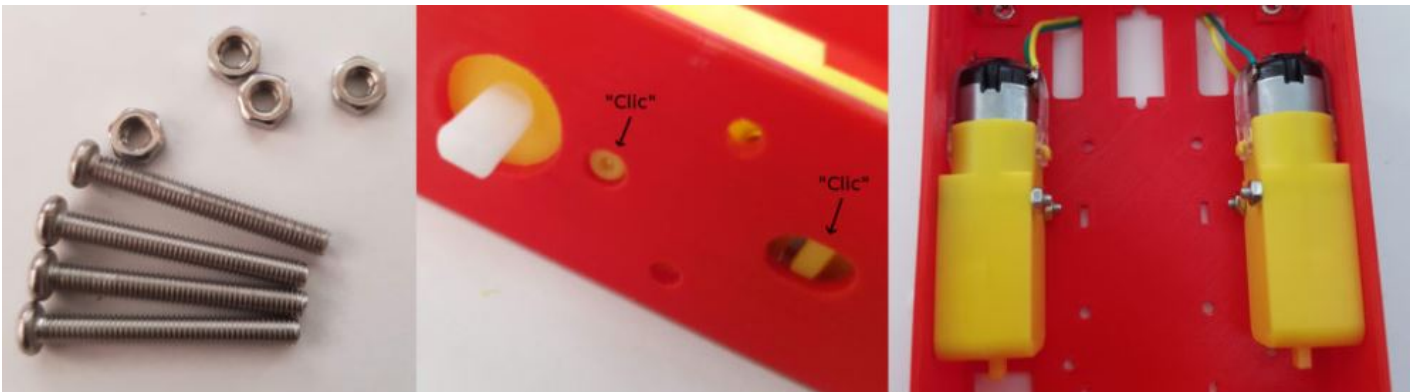
El sensor ultrasónico en el soporte, encaja justo para que no se caiga. Pon el soporte con dos tornillos en el chasis:



Fuente: <https://mclon.org> Maria L CC-BY-SA

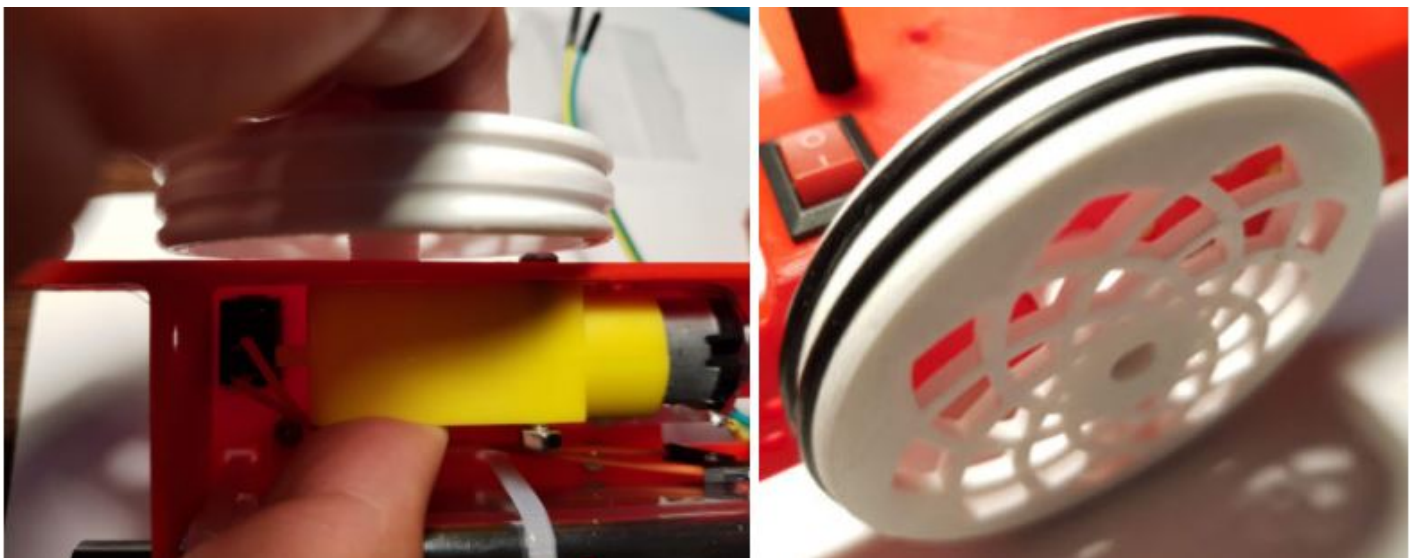
## 3.2.3 Motores y ruedas

Los motores, con 4 tornillos largos y pasa los cables por los orificios :



Fuente: <https://mclon.org> Maria L CC-BY-SA

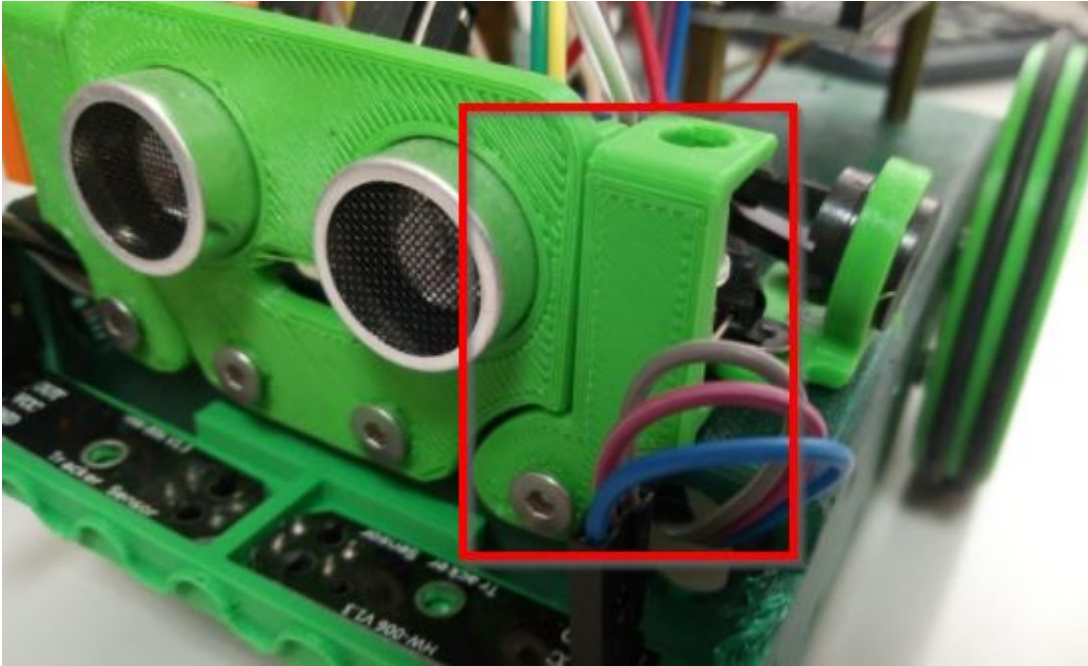
Pon las juntas tóricas en las ruedas y pasamos a su colocación: Las ruedas encajan muy justas, esto es así para evitar que con los golpes se desileneen. Hay que presionar con los dedos, si ves que cuesta puedes pasar una lima por el orificio para rebajarlo un poco. Fíjalo con los tornillos.



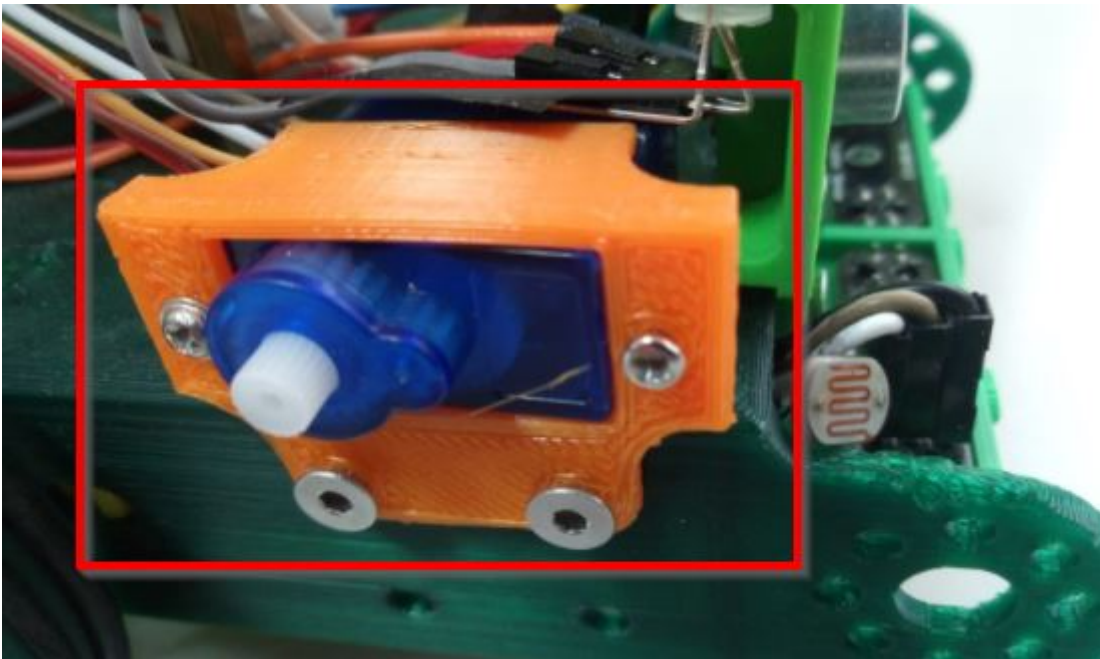
Fuente: <https://mclon.org> Maria L CC-BY-SA

## 3.2.4 placa y accesorios

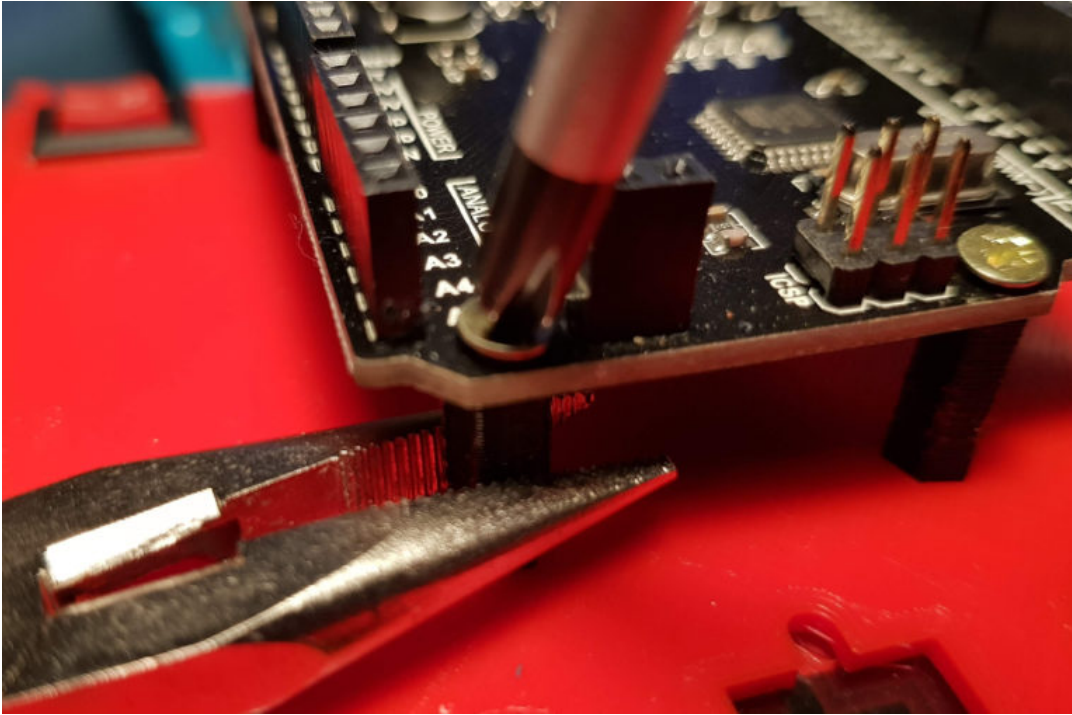
Pon los soportes de los leds RGB



Pon el soporte del servo al lado izquierdo (mirándolo a los ojos el ultrasonido)



Fija la placa al chasis



Fuente: <https://mclon.org> Maria L CC-BY-SA

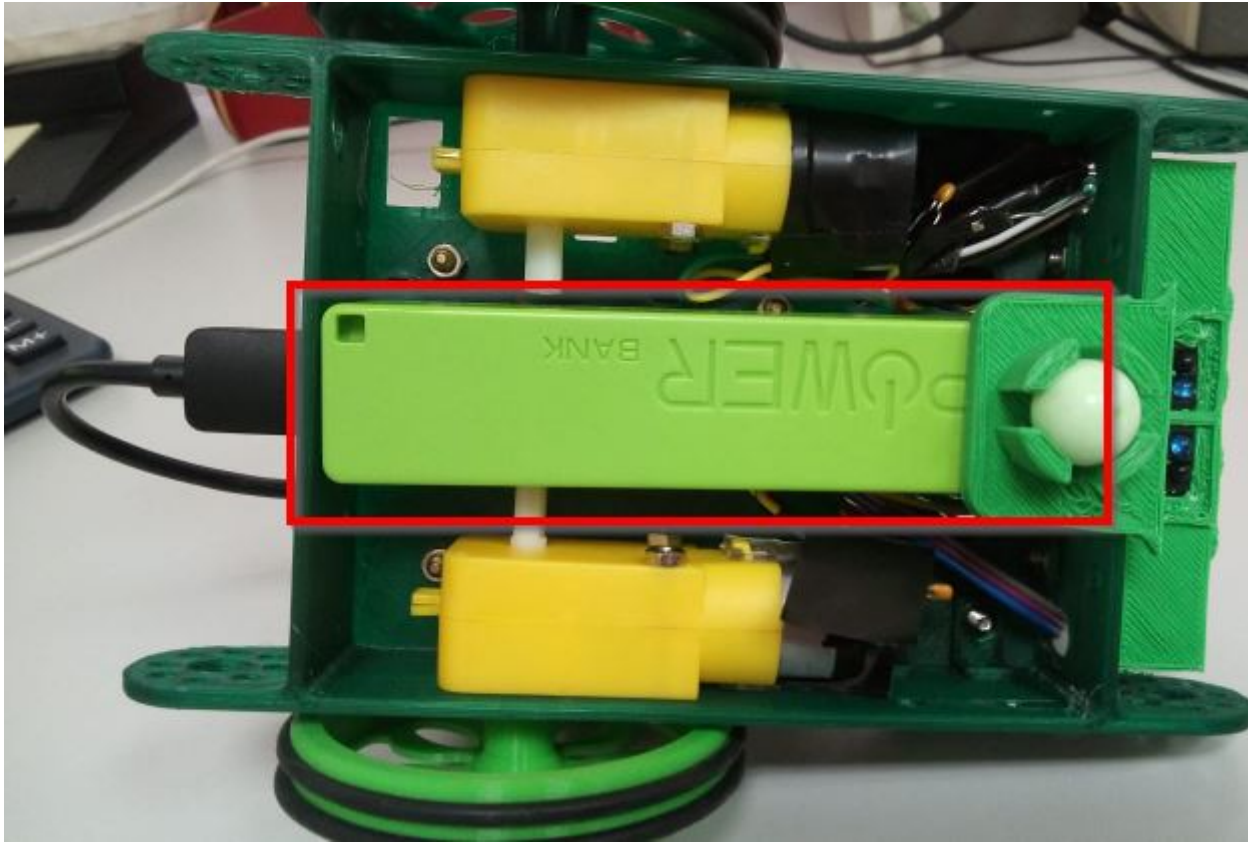
Fija el soporte del zumbador al lado derecho (mirándolo a los ojos el ultrasonido):



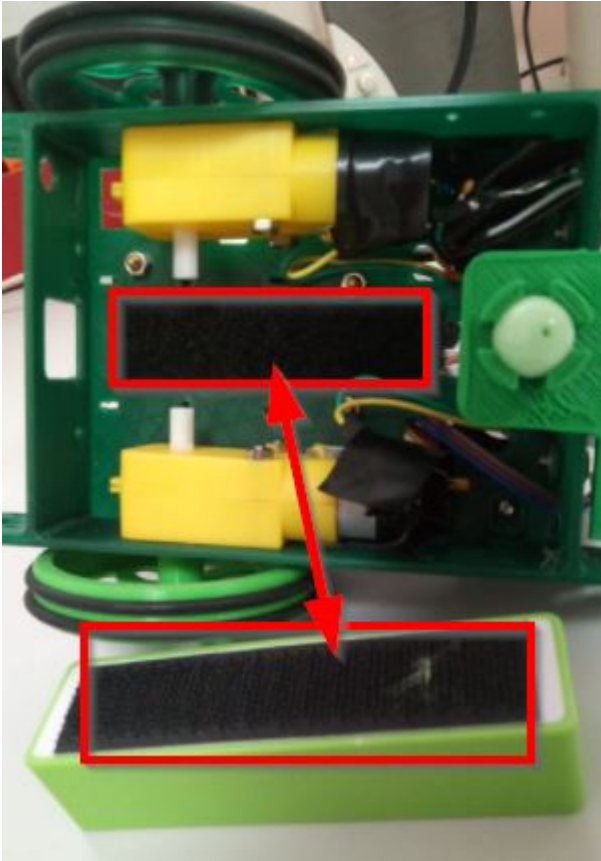
Fuente: <https://mclon.org> Maria L CC-BY-SA

## 3.2.5 Powerbank

El Powerbank se sujeta muy bien, pues en un extremo tiene el cable USB, en el otro extremo tiene la bola loca :



Y podemos poner un velcro para asegurarnos que no se mueva :



Esto permite sacarlo si fuese necesario, pero para tanto para la carga, conexión y desconexión del mClon no hace falta sacarlo nunca.

- Carga: Conectar el cable USB en el orificio microusb



- Conexión y desconexión del mClon por el puerto USB grande



3 Montaje

## 3.3 Cableado esquema

Este es la parte más difícil !!!

<https://giphy.com/embed/3o7abrH8o4HMgEAV9e>

[via GIPHY](#)

Tenemos que conseguir unir los diferentes elementos con los pines del Robodyn:

- Los pines digitales D0,D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13
- Los pines analógicos A0,A1,A2,A3,A4,A5,A6,A7
- Los pines de alimentación GND, VIN

Y tienen que ser estas conexiones y no otras para que sea compatible con mBlock y mBot [planos de mBot](#) :

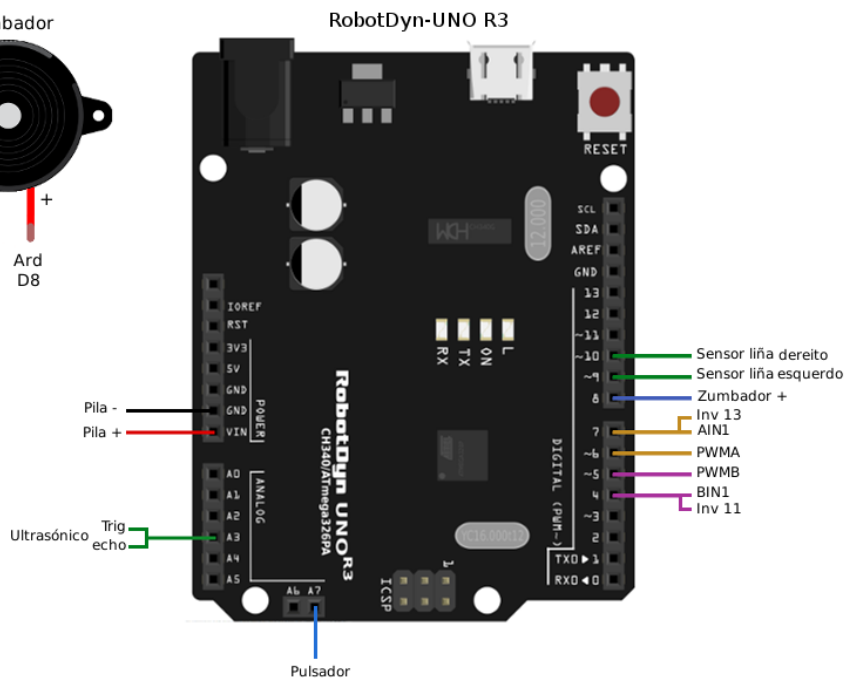
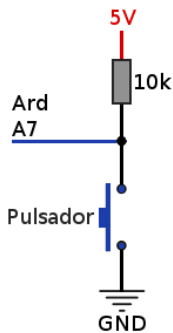
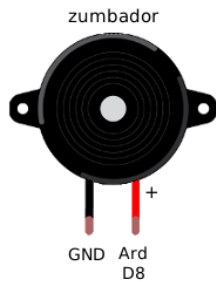
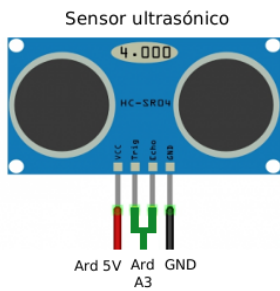
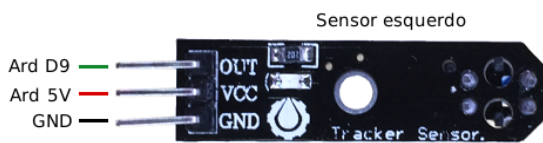
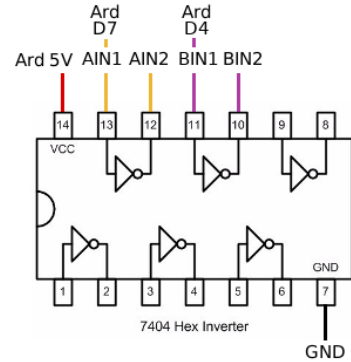
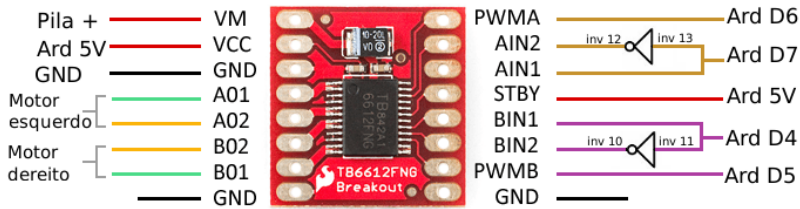
### 3.3.1.- Tabla de conexiones

PIN	ELEMENTO
A0	
A1	
A1	
A2	
A3	Pines Echo y Trg del sensor ultrasonidos
A4	
A5	
A6	AI LDR

PIN	ELEMENTO
A7	Pulsador
D1	
D2	
D3	
D4	Pin BIN1 del driver B6612FNG y en pin 11 del 7404
D5	Pin PWMB del driver B6612FNG
D6	Pin PWMA del driver B6612FNG
D7	Pin AIN1 del driver B6612FNG y en pin 13 del 7404
D8	Buzzer
D9	Sensor izquierdo siguelineas
D10	Sensor derecho siguelineas
D11	AI servo del brazo
D12	
D13	Leds RGB

### 3.3.2.- Esquema elementos básicos

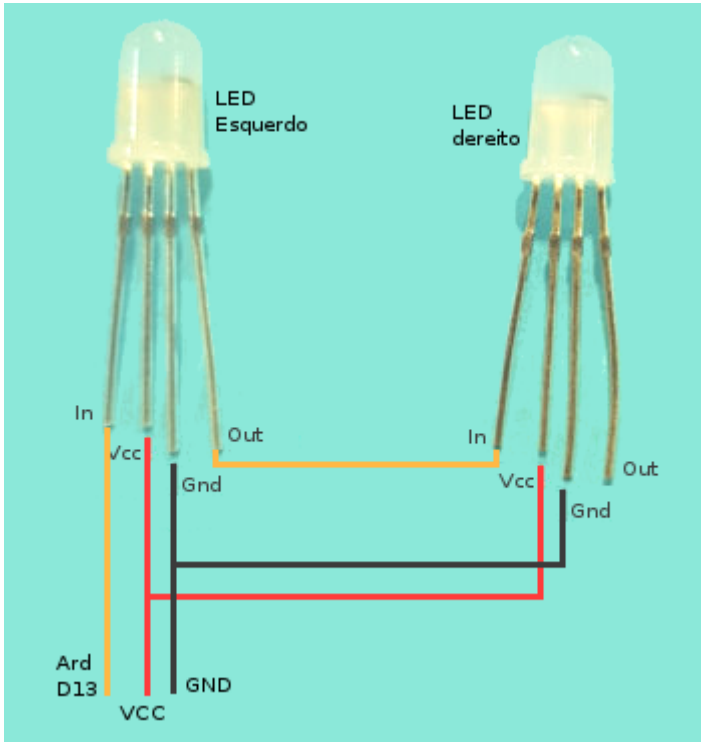
# mClon- esquema



Fuente: <https://mclon.org> Maria L CC-BY-SA

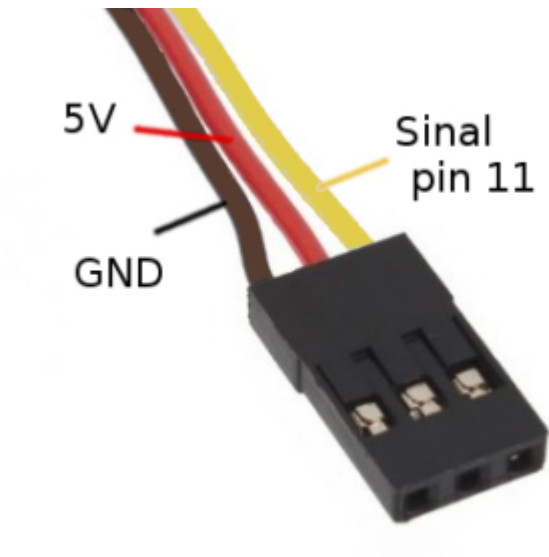
## 3.3.3 Esquema accesorios

Los dos leds RGB



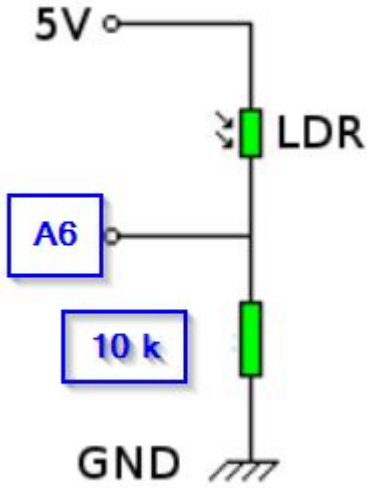
Fuente: <https://mclon.org> Maria L CC-BY-SA

El servo del brazo:



Fuente: <https://mclon.org> Maria L CC-BY-SA

El LDR :

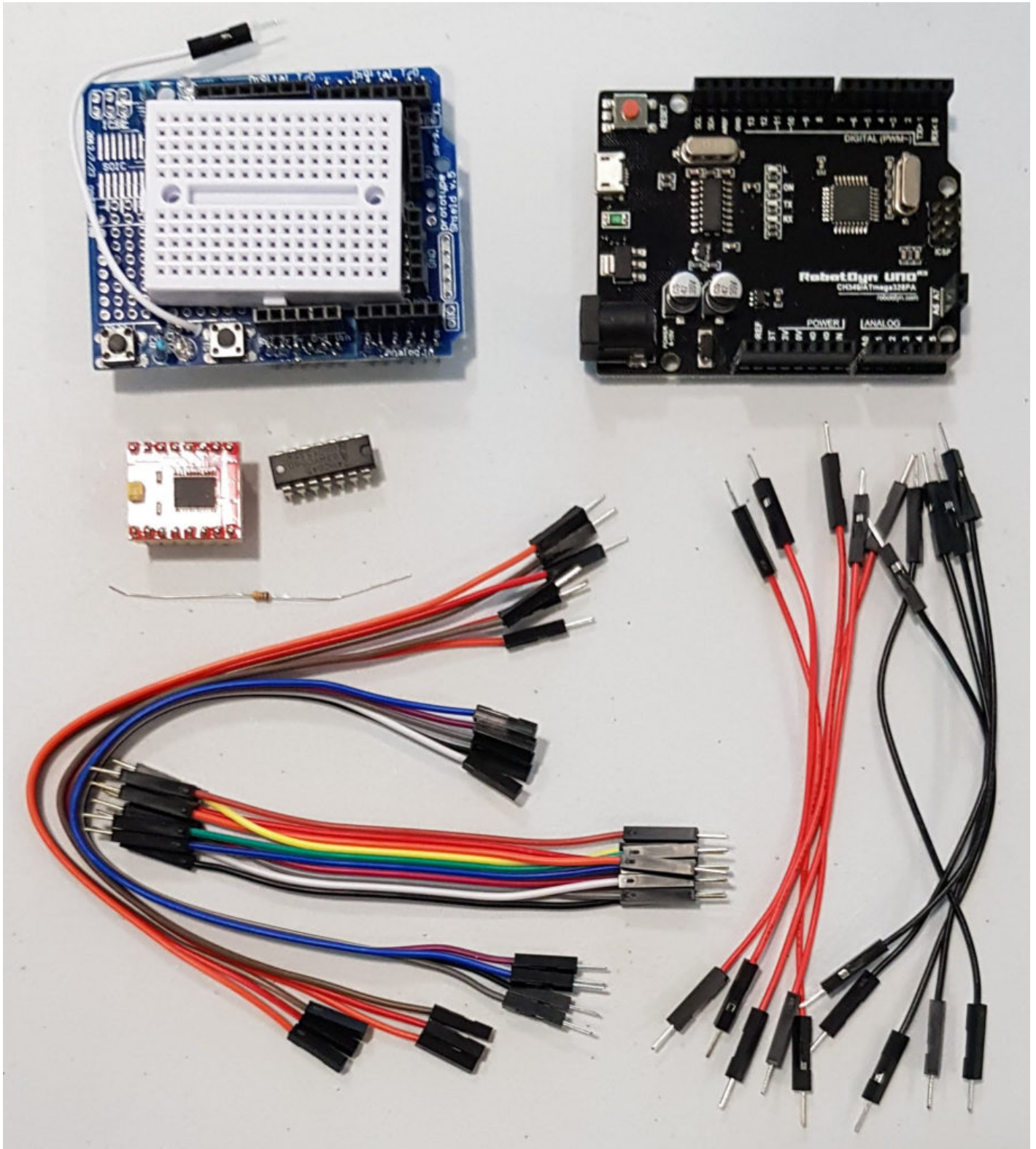


Fuente: Adaptado de <https://mclon.org> Maria L CC-BY-SA

3 Montaje

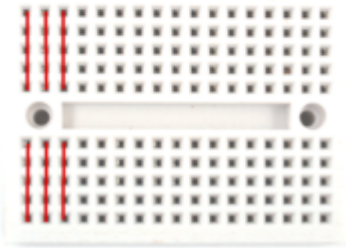
## 3.4 Cableado motores

Tenemos el Robodyn, el escudo Protoboard, cables, el driver motor B6612FNG, el 7404 y una R10k



Fuente: <https://mclon.org> Maria L CC-BY-SA

Recuerda que la placa protoboard, los agujeros están conectados verticalmente a ambos lados, es decir lo rojo está conectado:

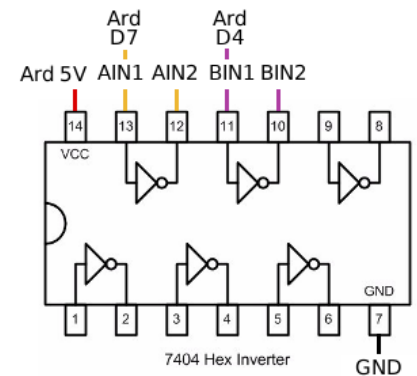
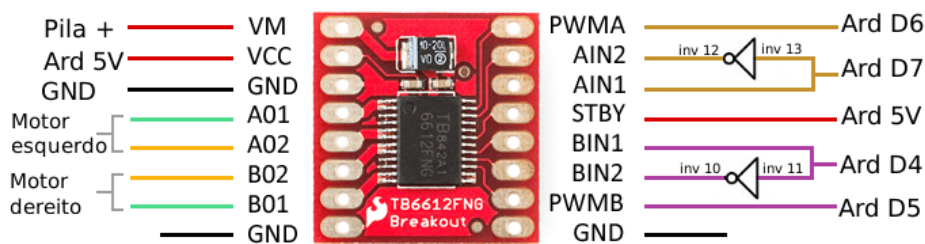


Fuente: <https://mclon.org> Maria L CC-BY-SA

En <https://tecnologia.org/mclon/electronica/protoshield/> tienes unos consejos para realizar correctamente las conexiones.

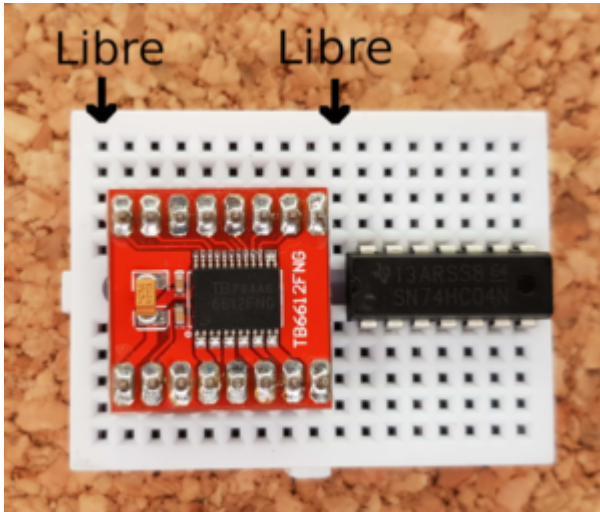
## 3.4.1 Driver motor B6612FNG y 7404

Vamos a conectar este esquema



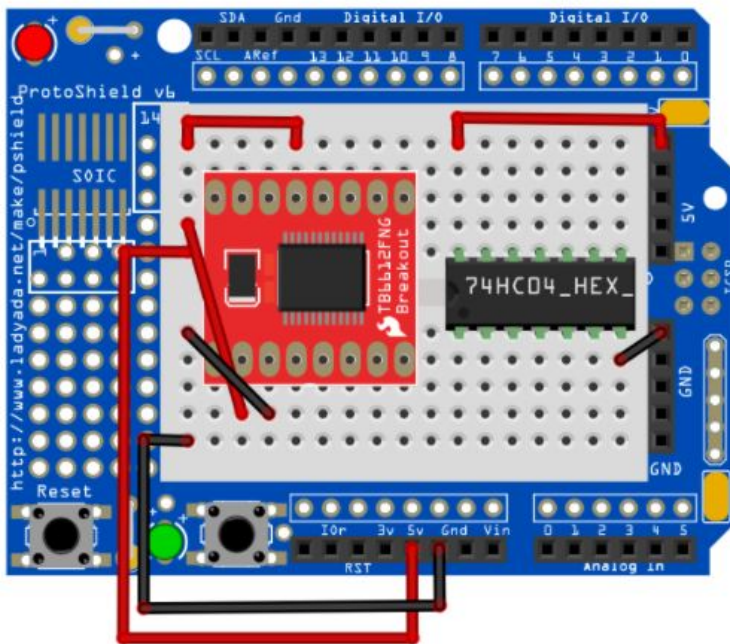
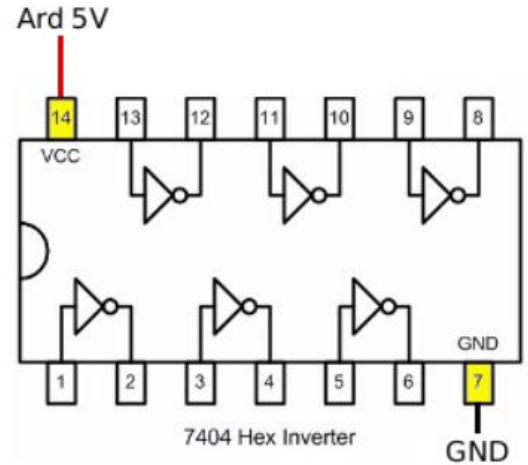
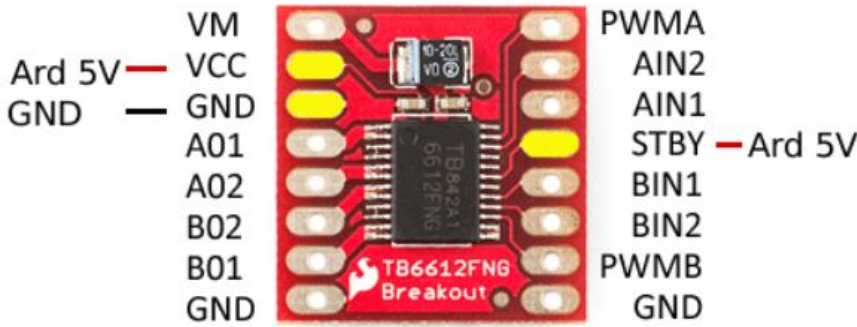
Fuente: <https://mclon.org> Maria L CC-BY-SA

Conecta los chips **dejando dos columnas libres** tal y como indica la figura, y conectamos Vcc y GND en la columna libre de la izquierda:



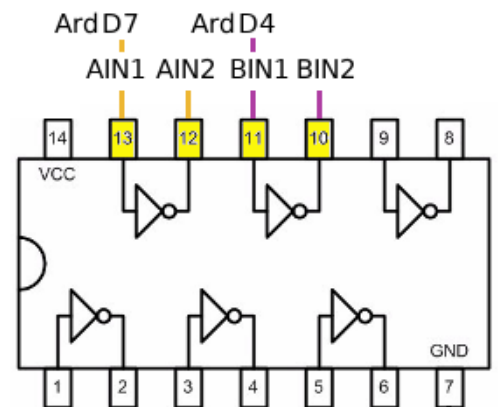
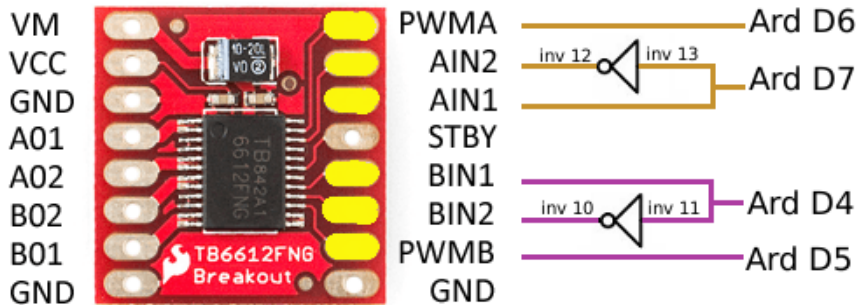
Fuente: <https://mclon.org> Maria L CC-BY-SA

Ponemos las conexiones de alimentación de los dos chips



Fuente: <https://mclon.org> Maria L CC-BY-SA

Y ahora conectamos los pines de velocidad y dirección:

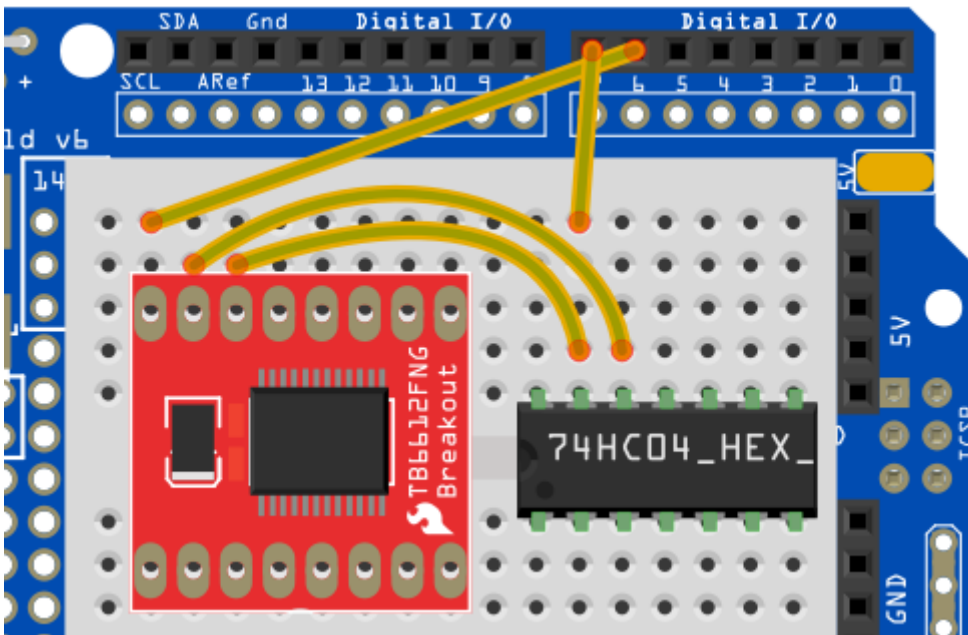


Fuente: <https://mclon.org> Maria L CC-BY-SA

Es decir, estos cables que indican en la siguiente figura:

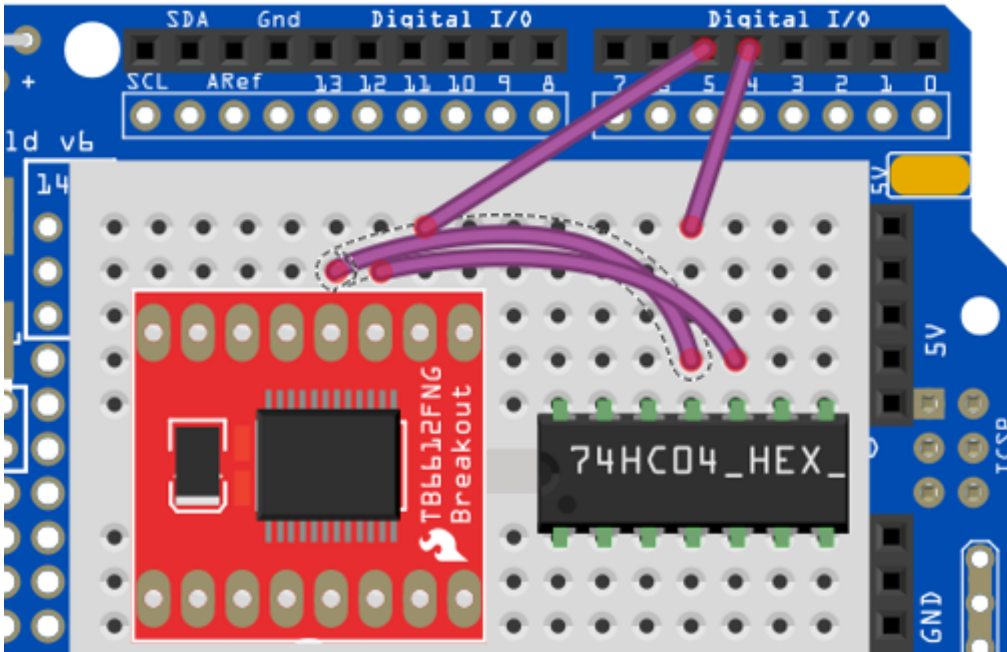
ATENCIÓN, no se han dibujado las conexiones anteriores, para simplificar los dibujos, es decir ,**no** quites los cables anteriores

Primero el motor A



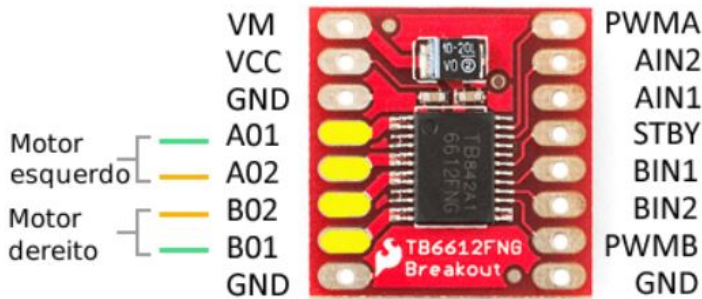
Fuente: <https://mclon.org> Maria L CC-BY-SA

Y luego el motor B



Fuente: <https://mclon.org> Maria L CC-BY-SA

Ahora conectamos los motores **OJO CON LA POLARIDAD** si te equivocas, el motor girará al revés :



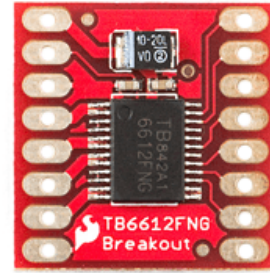
Al alimentar mClon con PowerBank, la tensión de la pila le llega directamente al Arduino Robodyn, luego alimentamos Vm con los 5V del Arduino (Vcc ya estaba a 5V)



Ao porwerbank

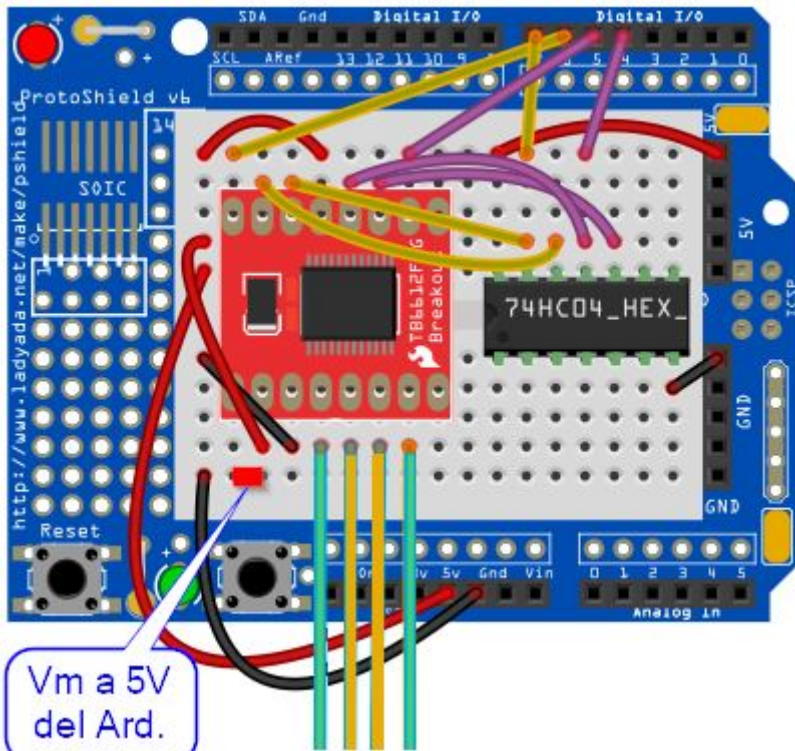
Á placa RobotDyn

- Ard 5V — VM
- Ard 5V — VCC
- GND
- A01
- A02
- B02
- B01
- GND



- PWMA
- AIN2
- AIN1
- STBY
- BIN1
- BIN2
- PWMB
- GND

Luego las conexiones quedan así :



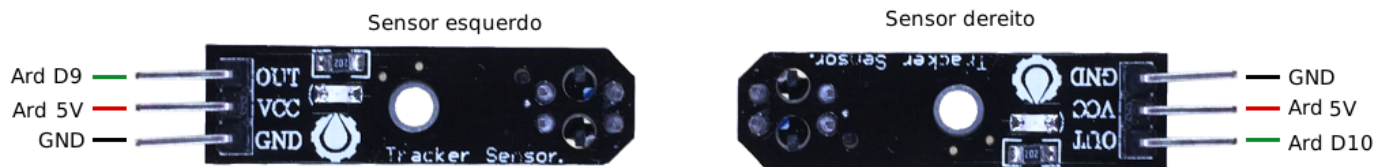
### 3 Montaje

## 3.5 Cableado sensores

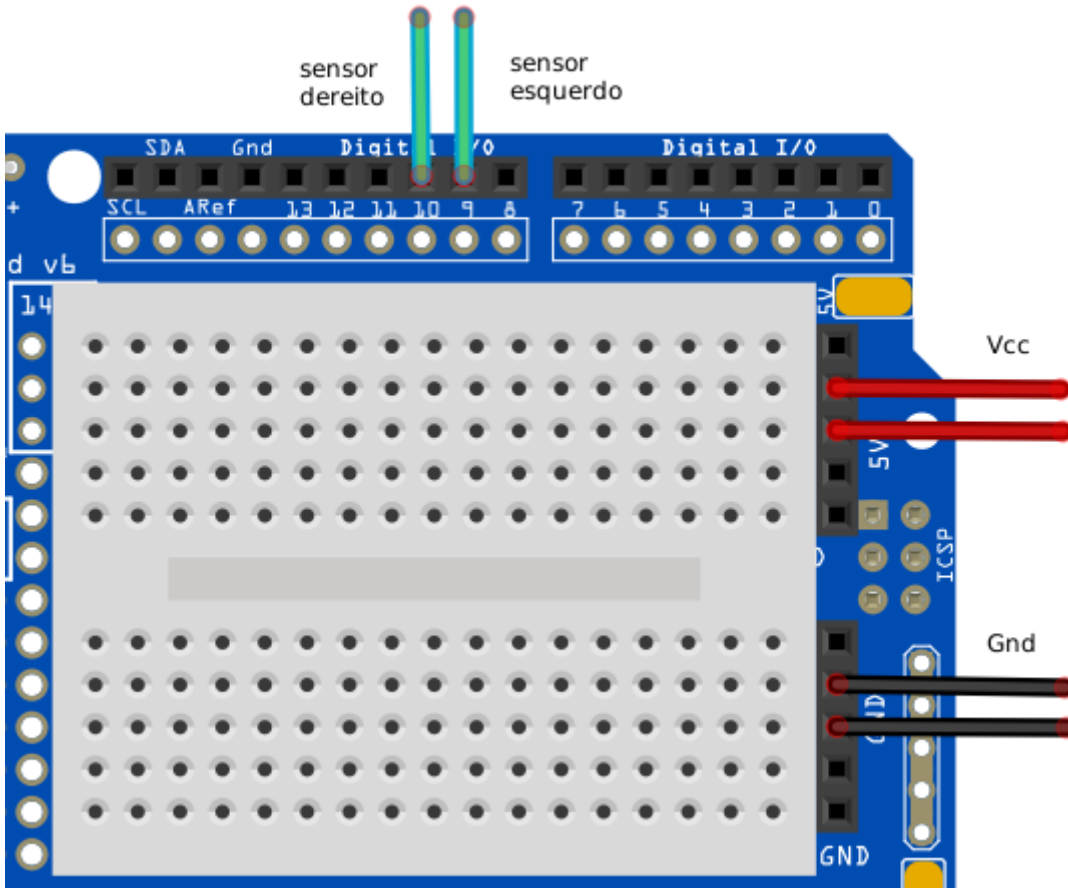
### 3.5.1 Sensores de línea

Se conlocan en el soporte de la cabeza loca, van justos, incluso puede ser no necesario atornillarlos. Tiene 3 pines:

- Vcc al 5V del Arduino.
- GND al GND del Arduino.
- OUT del sensor derecho a D9 mirándolo a los ojos del ultrasonido.
- OUT del sensor izquierdo a D10 mirándolo a los ojos del ultrasonido.



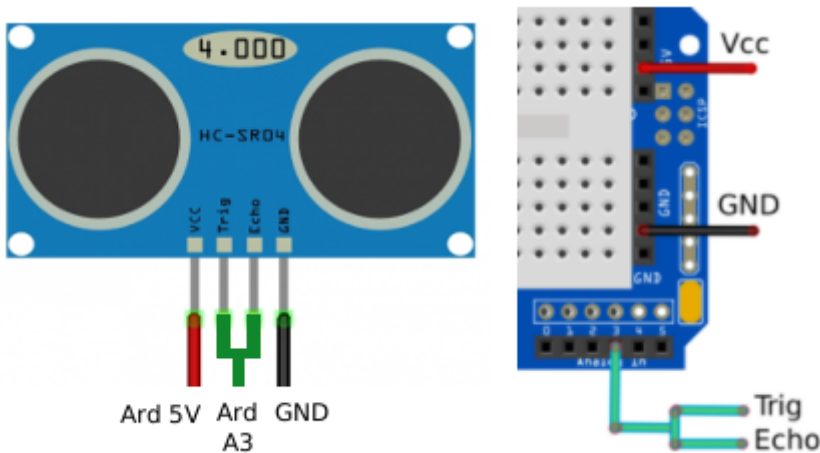
Fuente: <https://mclon.org> Maria L CC-BY-SA



Fuente: <https://mclon.org> Maria L CC-BY-SA

### 3.5.2 Sensor ultrasonidos

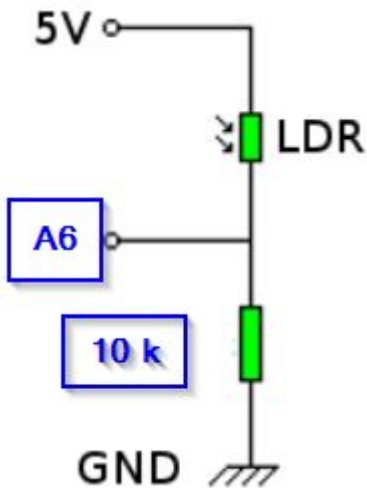
Los pines Trig y Echo del sensor **tienen que estar soldados** por lo que se conecta uno de los dos al A3



Fuente: <https://mclon.org> Maria L CC-BY-SA

## 3.5.3 Sensor de luz LDR.

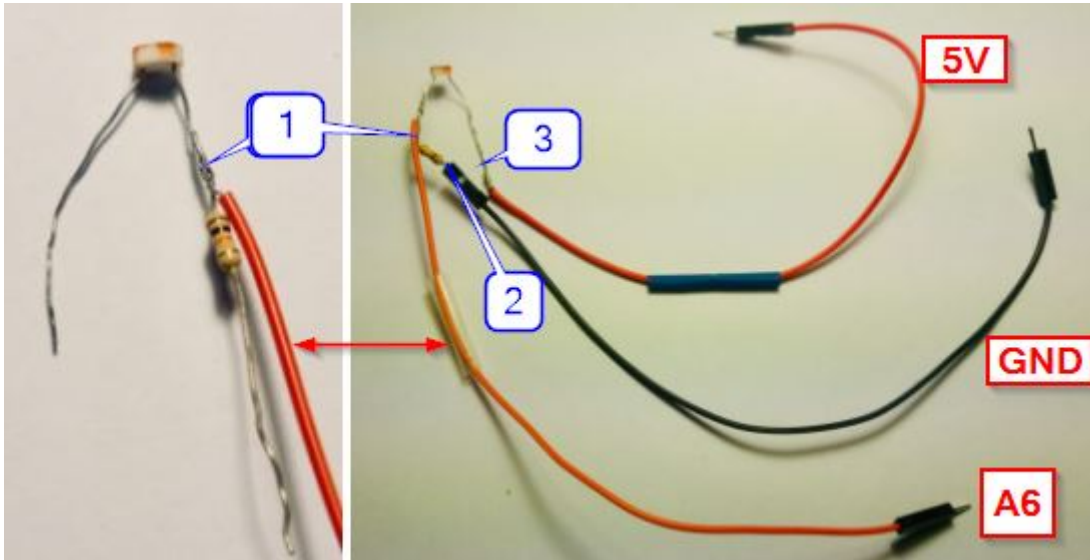
Para el sensor de luz, tenemos que hacer el siguiente esquema :



Fuente: Adaptado de <https://mclon.org> Maria L CC-BY-SA

Luego tenemos que hacer los siguientes empalmes, conectados o mejor soldados y aislarlos con una cinta aislante:

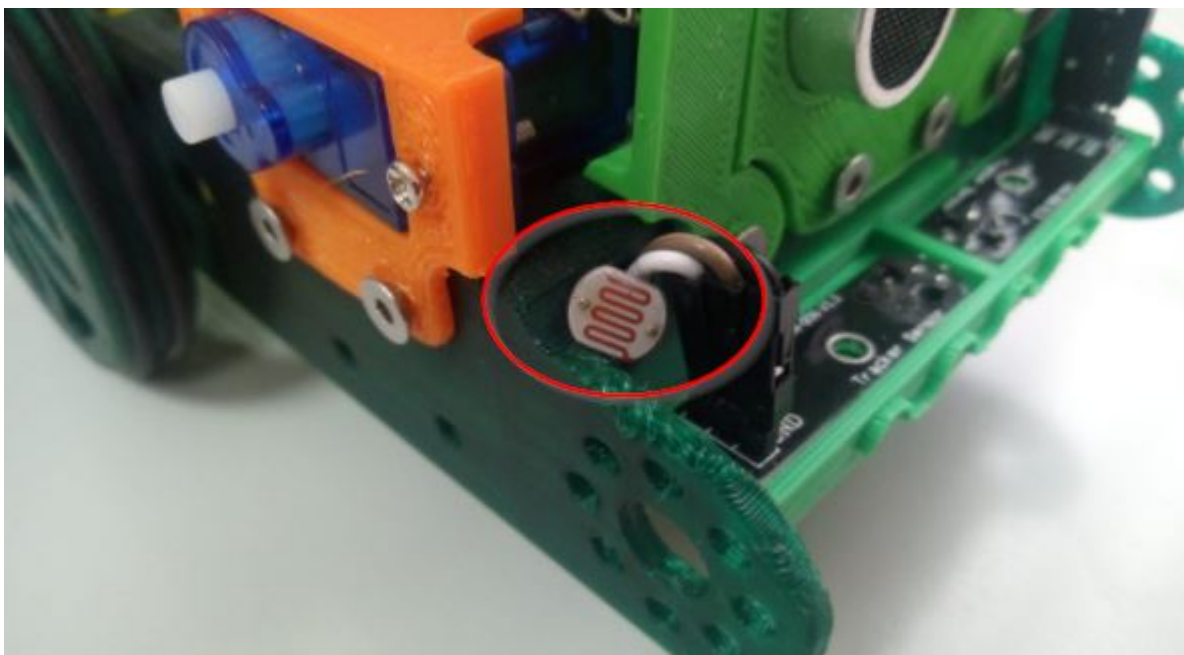
1. La resistencia 10k a uno de los pines del LDR y al cable que tiene que ir al A6
2. Un cable Dupond al otro extremo de la R10k que irá a GND del Arduino.
3. Un cable Dupond al otro extremo del LDR que irá al 5V del Arduino.



Fuente: Adaptado de <https://mclon.org> Maria L CC-BY-SA

Nuestra propuesta es de sólo **un LDR** pero recuerda que en [2.2](#) vimos una opción de poner dos LDRs con interesantes propuestas y fácilmente realizable.

Por simplificación, no se ha impreso en 3D ningún soporte especial LDR al chasis, pero en <https://tecnologia.org/mclon/estructura/impresion-3d/> lo tienes por si quieres imprimírtelo. Nosotros proponemos poner la resistencia en la parte debajo del chasis, pasar los cables hacia el escudo Protoboard por los numerosos agujeros que hay en el chasis, y pasar el LDR por el agujero de los cables del sensor siguelíneas :

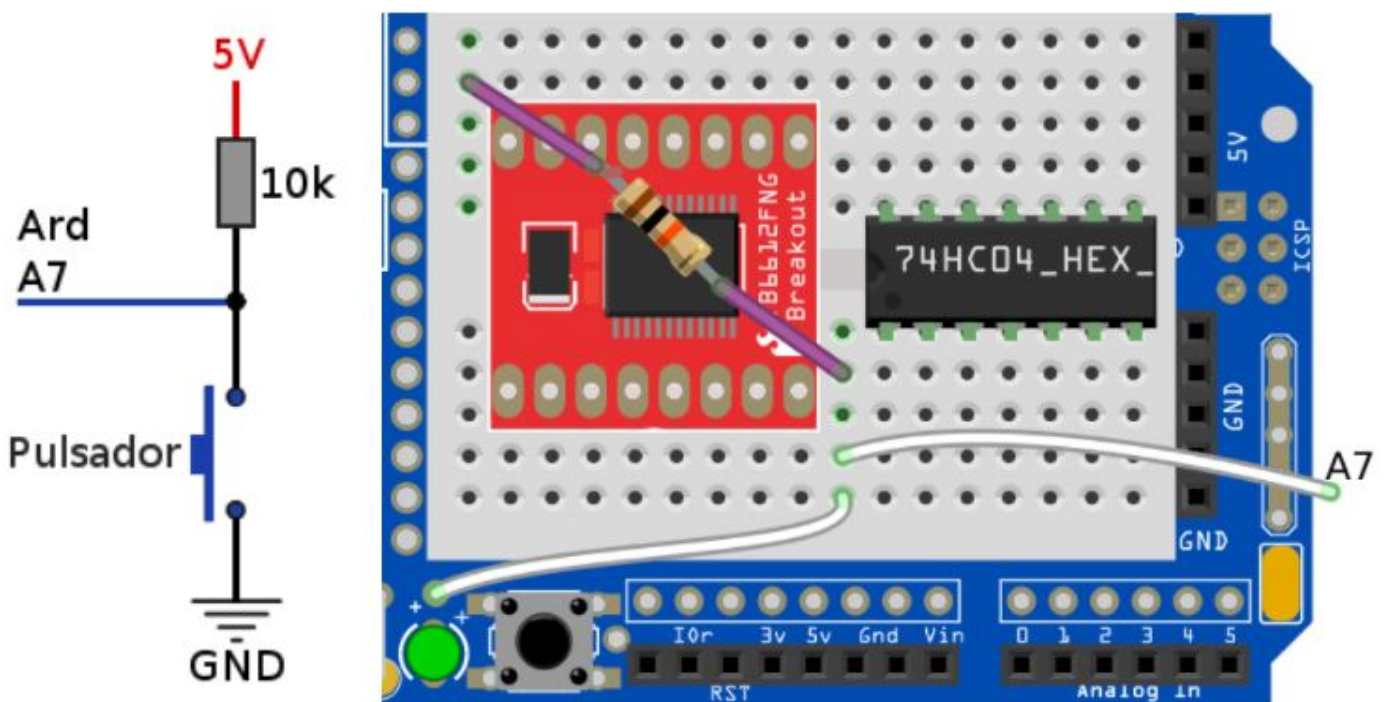


3 Montaje

## 3.6 Cableado accesorios

### 3.6.1 Pulsador

El pulsador lo tenemos que conectar al A7 que no tiene Arduino, por eso nuestra propuesta es con Robotyn, y para ello utilizaremos la segunda columna que dejamos libre en el Protoboard :



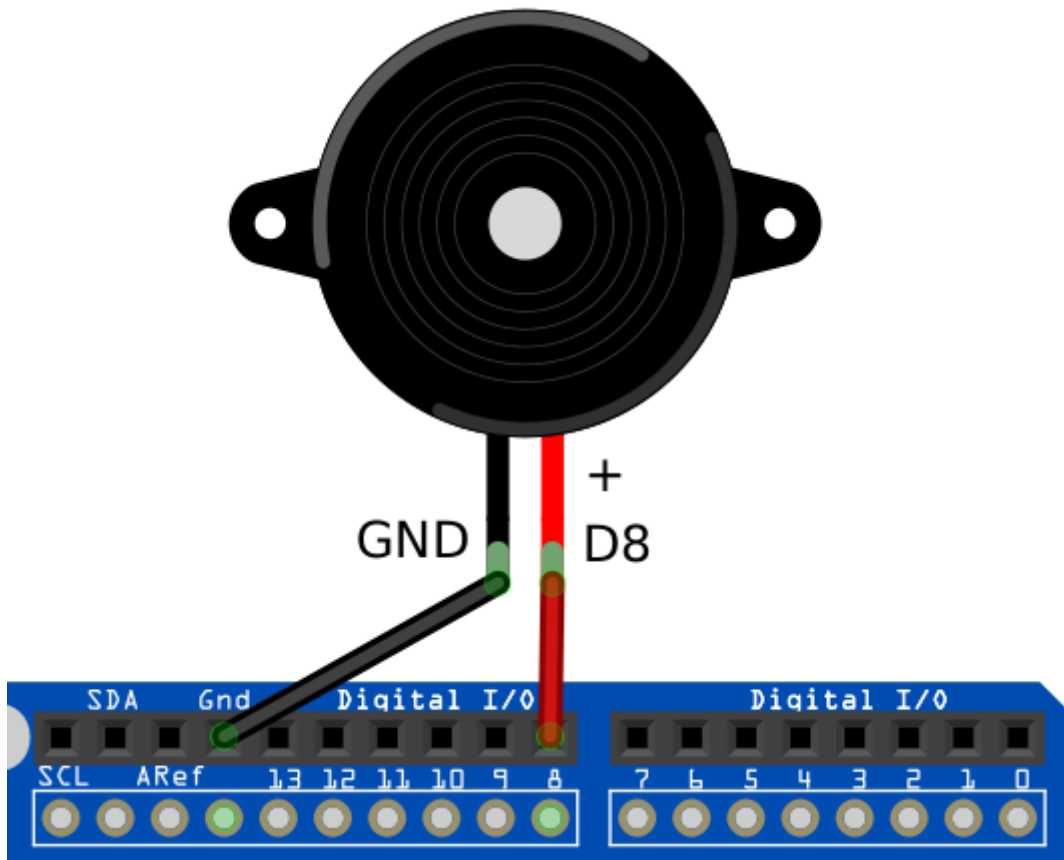
Fuente: <https://mclon.org> Maria L CC-BY-SA

Recuerda que el A7 lo dejastes preparado en [3.1 Pasos previos](#)

**Evita** que los pines de la resistencia 10K toquen los pines del controlador, puedes poner una cinta aislante.

### 3.6.2 El zumbador

Es sencillo, simplemente conéctalo a D8 y el otro a masa :



Fuente: <https://mclon.org> Maria L CC-BY-SA

3 Montaje

## 3.7 Cableado extras

Vamos a utilizar los famosos **LED NEOPIXEL**, el origen de este nombre es de la compañía ADAFRUIT y son leds de tres colores, que dentro tienen un integrado que permite mandar secuencialmente órdenes a los siguientes, con lo que puedes hacer tiras de leds tan largo como desees y con multitud de posibilidades de juegos de luces disponibles.

En el mClon sólo vamos a trabajar con dos leds, uno recibirá el orden del Arduino por su pata IN,, y se lo comunicará al otro led por su pata OUT al IN del segundo led, y el segundo led, su pata OUT se queda sin conectar pero podríamos seguir la secuencia tanto como quisiéramos.

Si quieres saber más te recomendamos la página de [e-lka](#) y la de [Luis Llamas](#)

### 3.7.1 Aclaración sobre estos leds RGB

Tienes **dos tipos FIJATE cual tienes pues el patillaje cambia**

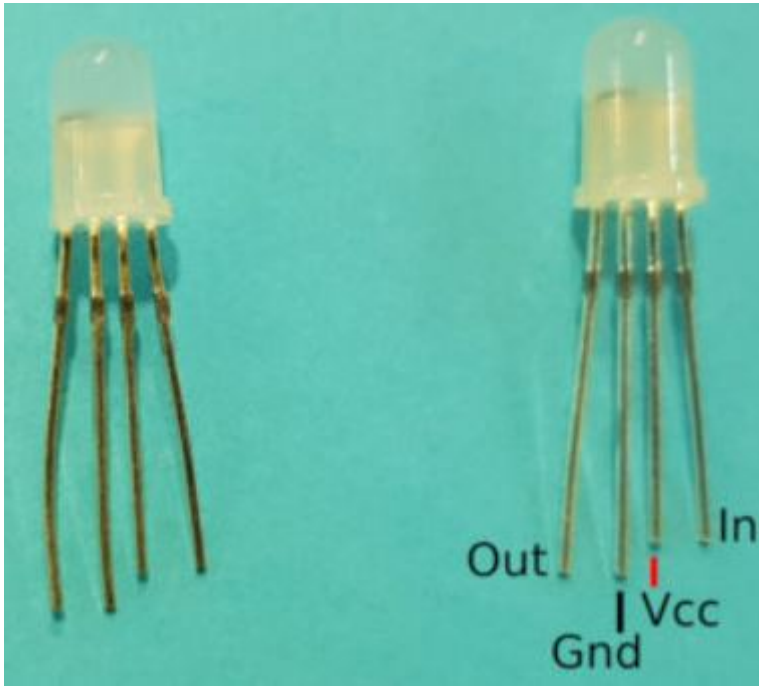
- APA 106-F5 de 5mm
- APA 106-F8 de 8mm

Realmente NO SON RGB SINO SON **GRB**, luego salen cambiados, el ROJO con el VERDE es decir, cuando pides rojo, sale verde y cuando pides verde sale rojo, en el azul está bien

En <https://tecnoloxia.org/mclon/extras/luces-de-cores/> tienes una explicación de todos los valores a utilizar y los colores que se obtienen.

#### 3.7.1.1 APA - F5 de 5mm

Es lo más recomendable, estos leds RGB tienen esta configuración :



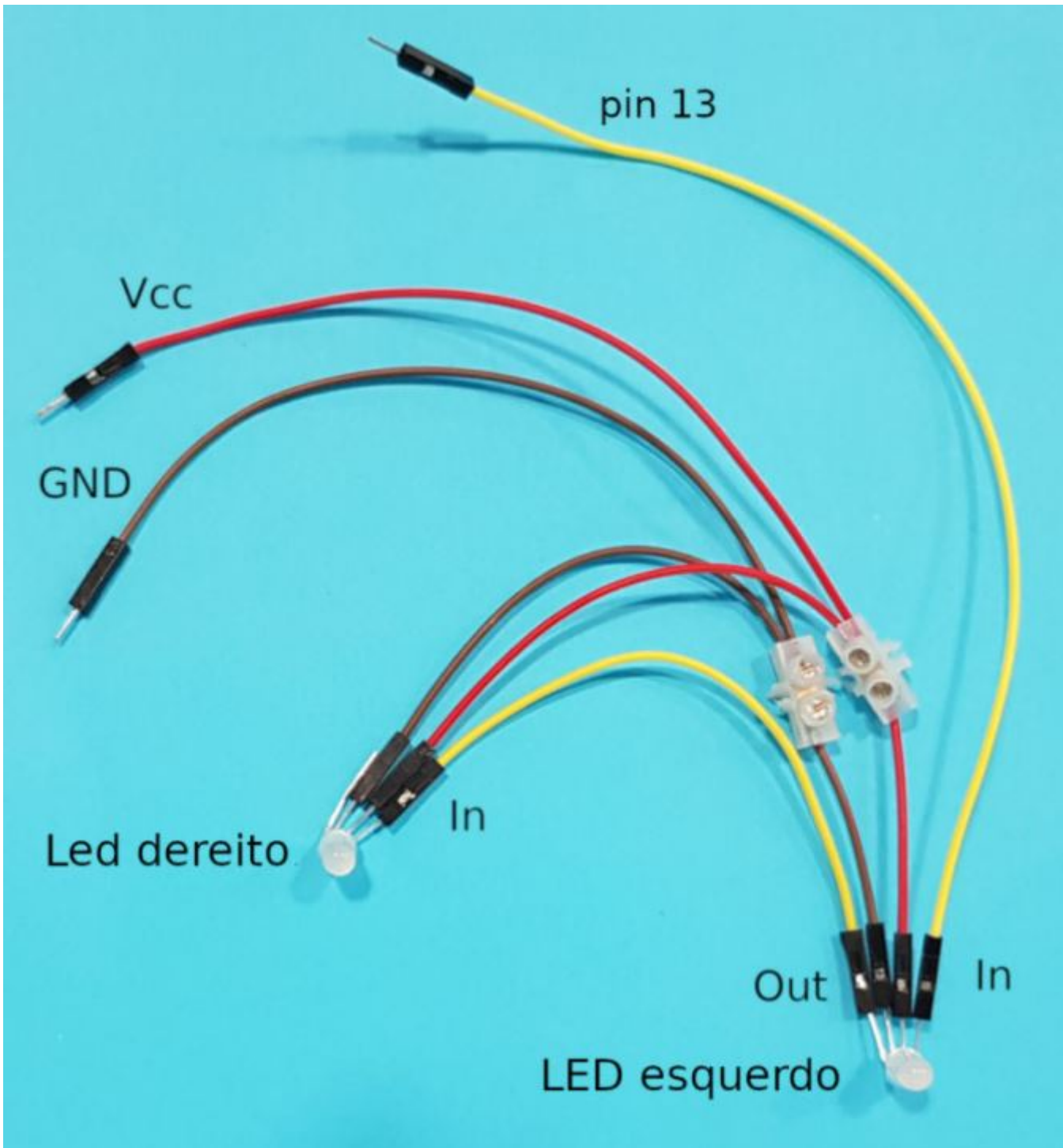
Fuente: <https://mclon.org> Maria L CC-BY-SA



Fuente: Datasheet

Con cables Dupond y con ayuda de regletas, podemos hacer la siguiente conexiones :

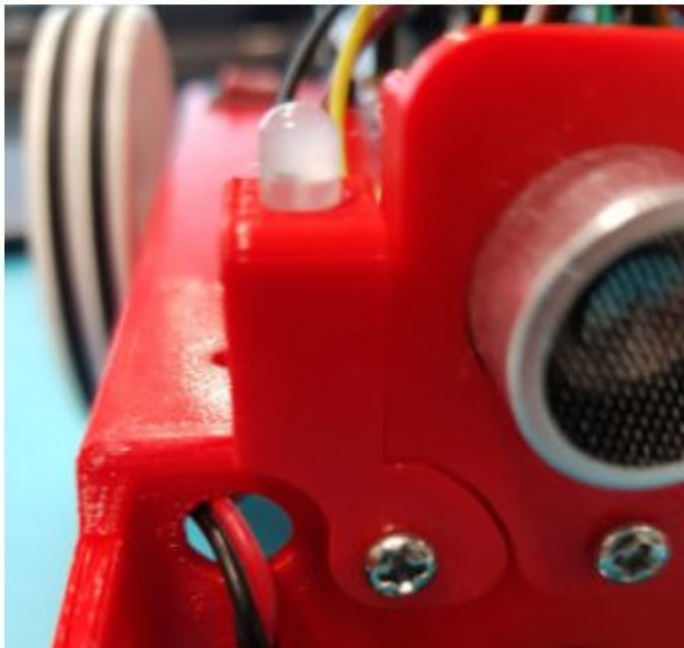
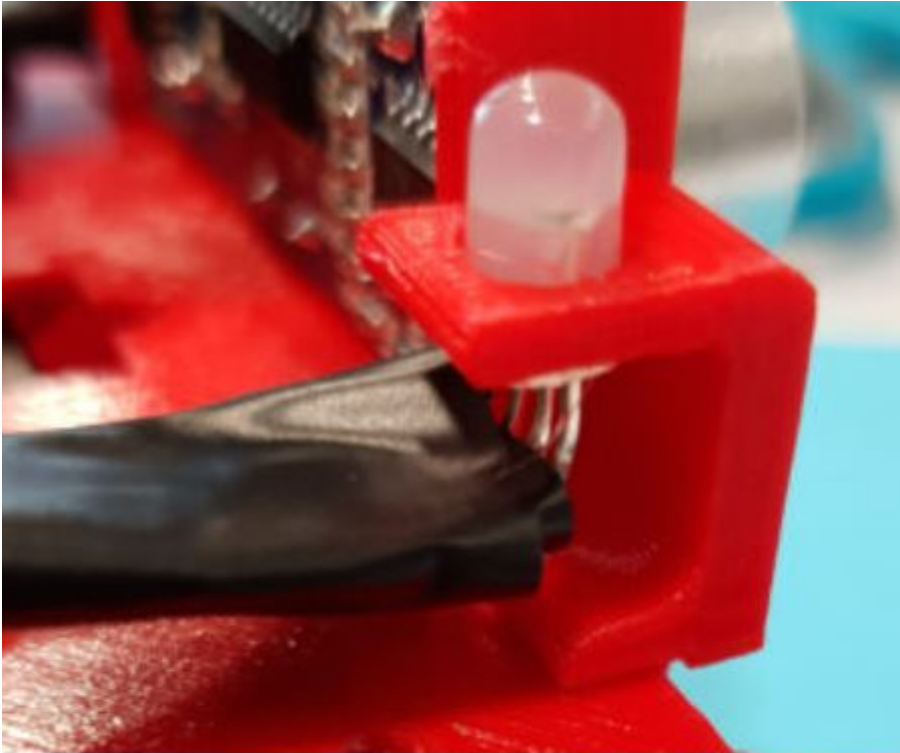
- Los dos Vcc conectados y al Vcc de la placa 5V
- Los dos GND conectados y al GND de la placa 0V
- El Din de un led al pin 13 de la placa Arduino
- El Dout de ese led al Din del otro led
- El Din del otro led sin conectar



Fuente: <https://mclon.org> Maria L CC-BY-SA

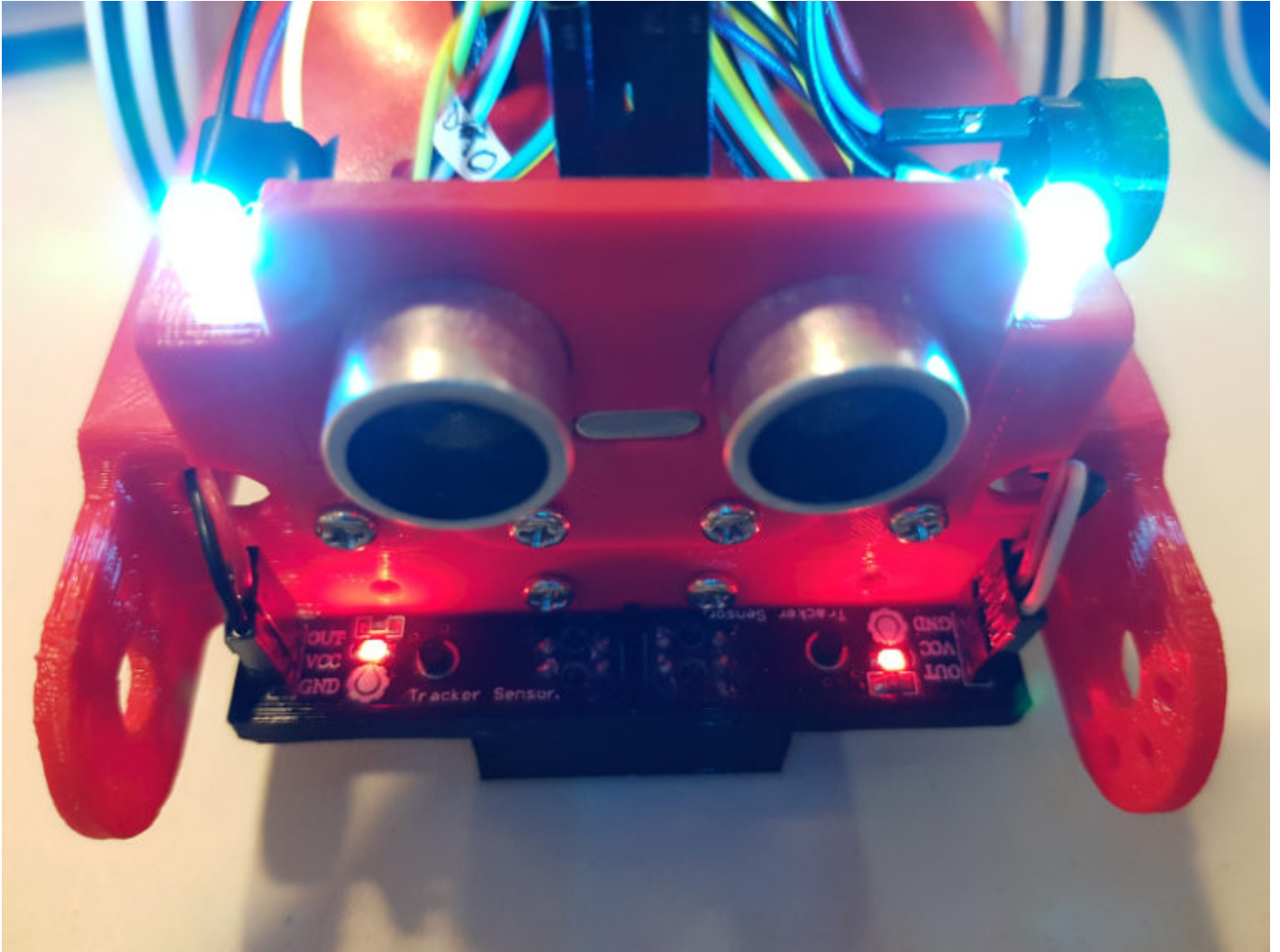
*Sujeta las conexiones de los cables Dupond y los leds RGB con cinta aislante para que no se desconecten.*

Y los colocas en el chasis en sus soportes :



Fuente: <https://mclon.org> Maria L CC-BY-SA

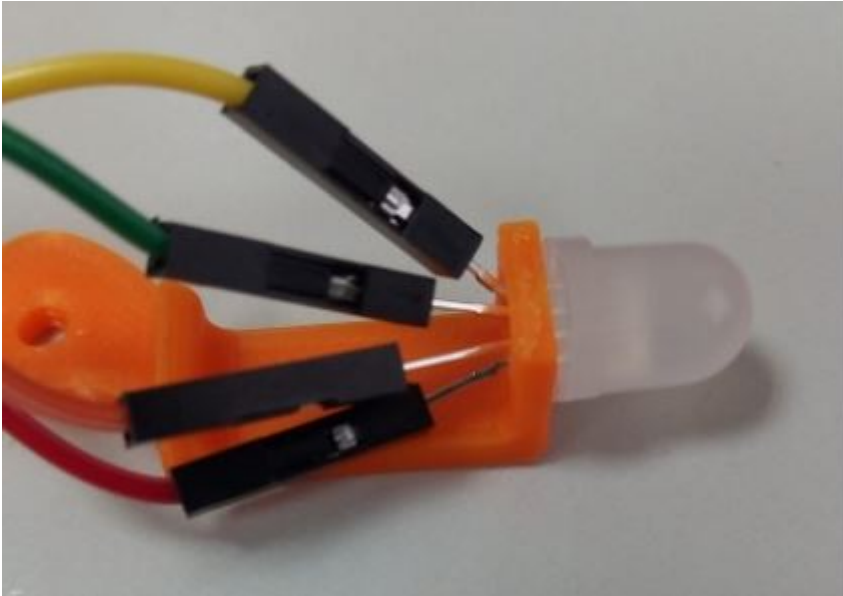
Conecta el cable rojo a 5V, el negro a GND y el amarillo al pin 13, y voila !!



Fuente: <https://mclon.org> Maria L CC-BY-SA

### 3.7.1.2 APA - F8 de 8mm

En este caso, el led no cabe por el hueco, tienes que ponerlo por encima:

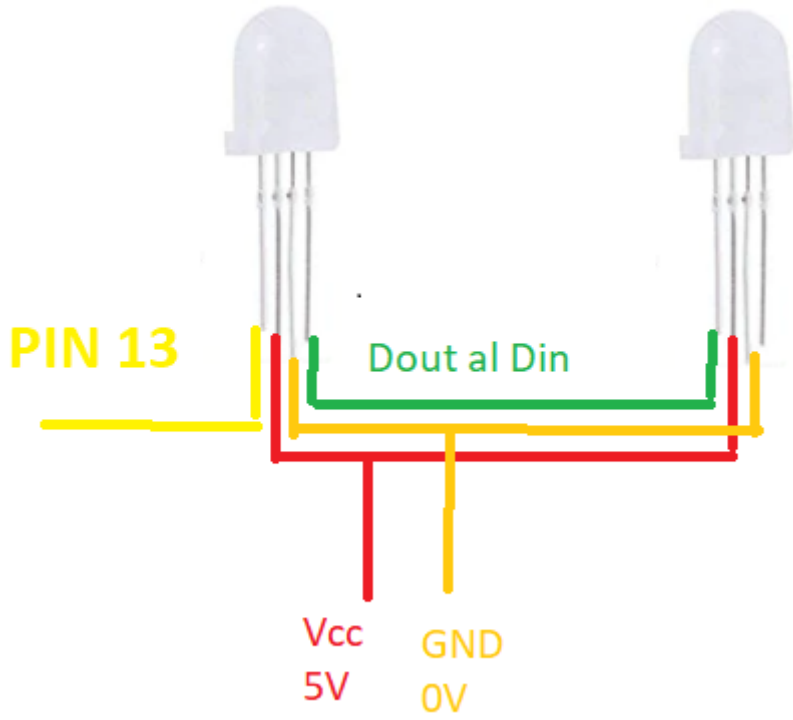


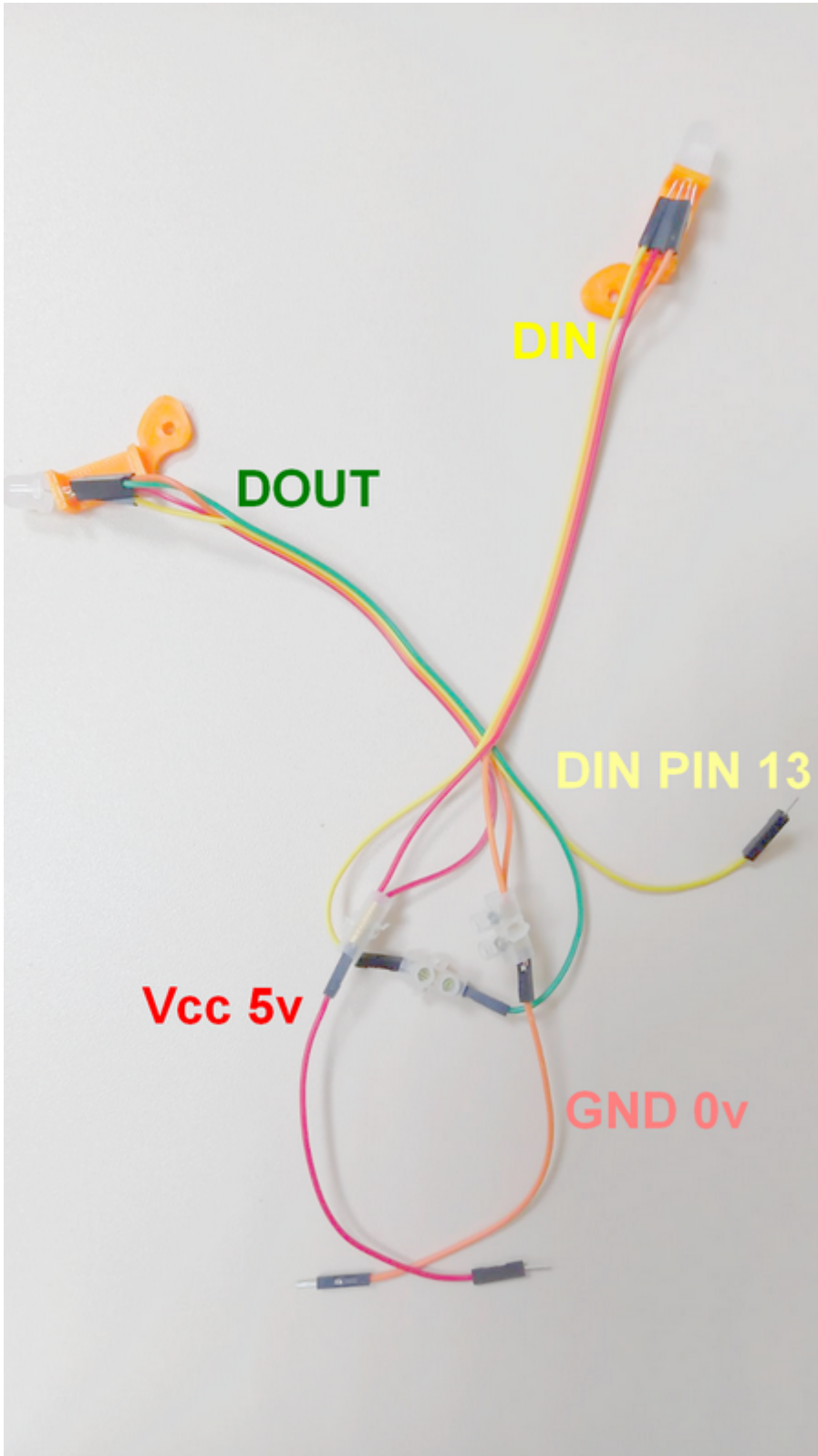
Y su configuración de pines **es diferente** al APA-106-F5 :



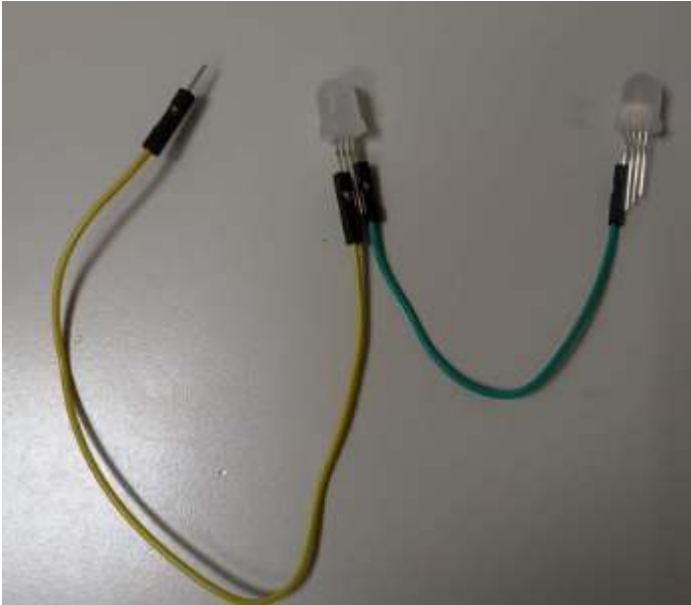
Luego la conexión es :

- Los dos Vcc conectados y al Vcc de la placa 5V
- Los dos GND conectados y al GND de la placa 0V
- El Din de un led al pin 13 de la placa Arduino
- El Dout de ese led al Din del otro led
- El Din del otro led sin conectar





Las regletas nos las podemos ahorrar utilizando diferentes agujeros de la placa Protoboard y la conexión Din-Dout entre los leds con un cable Dupont H-H, en la foto se omiten los cables +5V y GND por simplificar

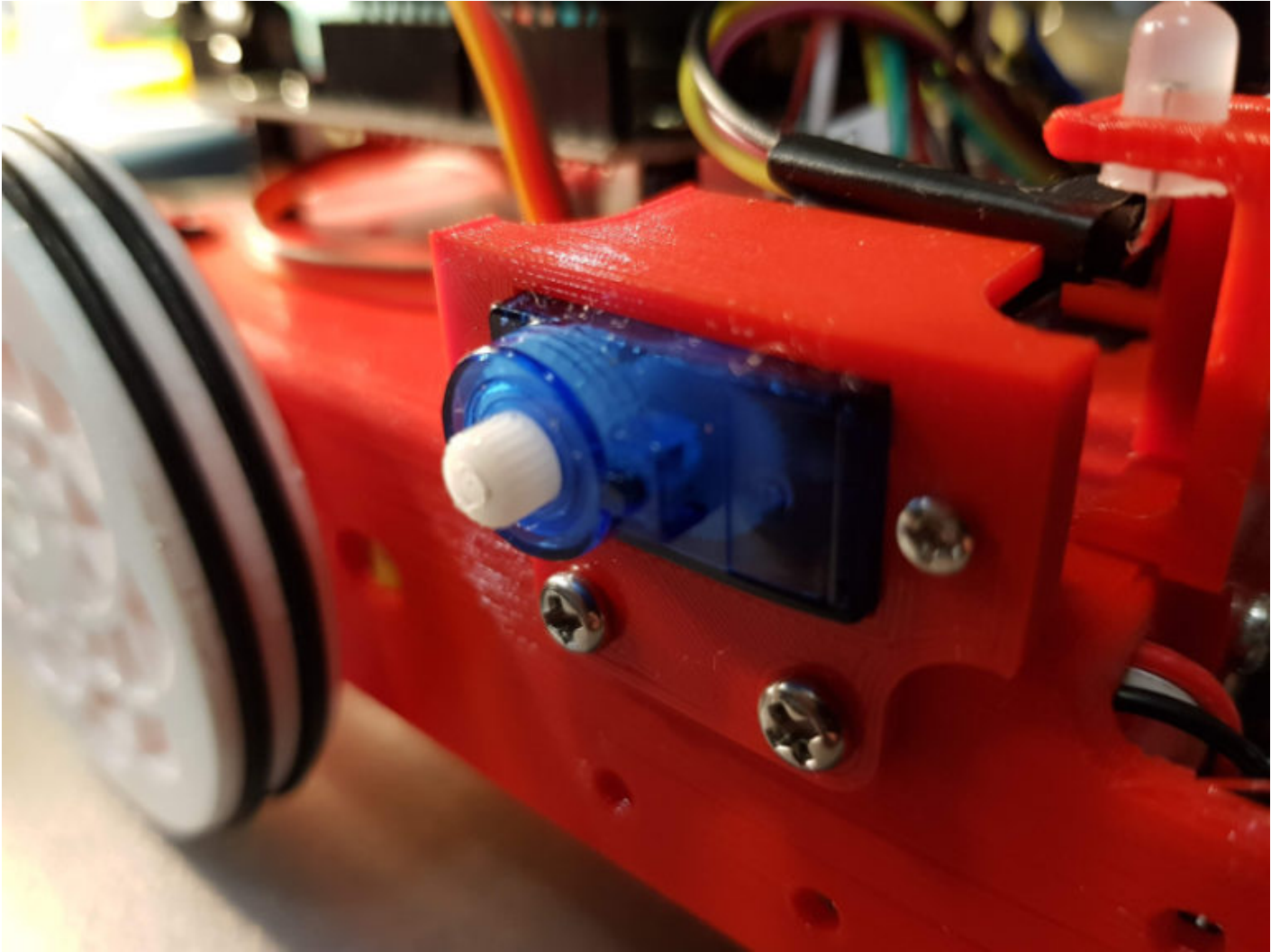


Una vez conectado, para que los cables Dupont no se salgan, puedes poner una cinta de celo



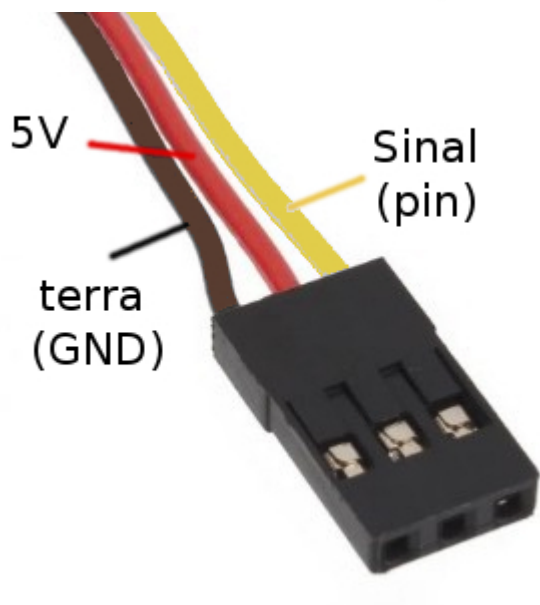
### 3.7.2 Brazo robot: Servomotor

Coloca el servomotor en su soporte usando los tornillos que vienen en su bolsa:



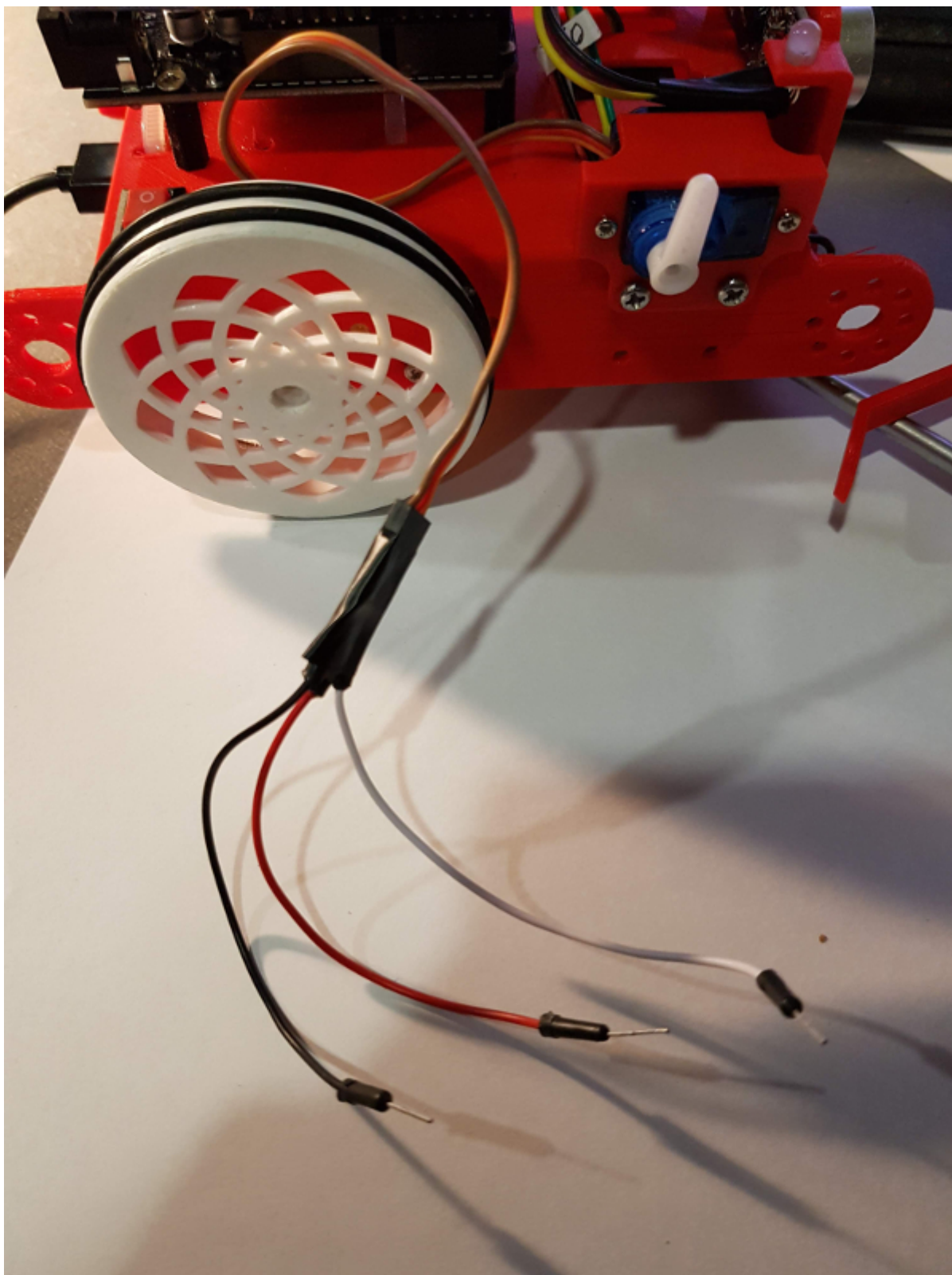
Fuente: <https://mclon.org> Maria L CC-BY-SA

Normalmente los colores de los cables del servo tienen este significado :



Fuente: <https://mclon.org> Maria L CC-BY-SA

Lo uniremos con 3 cables Dupond Macho-Macho y lo fijaremos con cinta aislante :



Fuente: <https://mclon.org> Maria L CC-BY-SA

Y conectaremos el cable amarillo Señal **al pin D11** para poder usar la instrucción (si queremos que sea slot2 o banco2 ponerlo a D12):



Finalmente colocamos el brazo, en el capítulo 4.5 lo calibraremos de forma que

- El ángulo de 0º corresponde a posición horizontal
- El ángulo de 90º corresponde a posición vertical

Una vez identificado cual es el ángulo 90º ejecutando la instrucción anterior, y una vez puesto el servo en esa posición, poner el brazo levantado con un ligero ángulo hacia delante, tal y como indica la figura :



Fuente: Adaptado de <https://mclon.org> Maria L CC-BY-SA

*Puedes poner un tornillo en para asegurarlo o no ponerlo para quitar el brazo y ponerlo con facilidad para que sea más cómodo el almacenaje del robot.*

# 4 Testeo

#### 4 Testeo

## 4.1 Con mBlock

Este tutorial no es un tutorial de programación de **mBlock** sólo se muestran algunas líneas principales de recordatorio. Recomendamos visitar los tutoriales de mBot en [www.catedu.es](http://www.catedu.es) en Robótica los correspondiente a mBot.


Para entrar en el programa mBlock tienes que entrar en <http://www.mblock.cc/> en Download y tienes varias opciones:

1. **Online** versión Web tiene la ventaja de tener tus programas en la nube si te registras pero dependes de la conexión de Internet en su uso en el aula.  
Hay que descargar previamente el driver **mLink** y ejecutarlo,. Lo encontrarás en la misma página
2. **Offline** descargando el programa en <http://www.mblock.cc/> en Download
  1. Para equipos viejos, incluso con Windows XP recomendamos la versión 3
  2. Para PC, la versión 5 tienes varias opciones en Windows, Mac o Tablets. En Linux usa la versión web.




# Download mBlock

One-stop coding platform tailored to coding education, trusted by 15 million educators, and learners

 **mBlock web version**  
 Chrome browser recommended >>  
 Support Windows/Mac/Linux/Chromebook

[Code with blocks](#) [Code with Python](#)

 **mBlock PC version**  
 Version: V5.4.0  
 Released: 2021.11.30  
[Released log >>](#) [Previous version >>](#)

[Download for Windows](#) [Download for Mac](#)  
Win7 or Win10 (64-bit recommended) macOS 10.12+

 **mBlock mobile app**  
 Learn coding in phones and tablets

 **Android**  
 Android 6.0 +  
 (ARM-based devices only. X86 Android not supported)

 **iOS**  
 iOS 10.0 +

## Other mBlock software

**mLink** mBlock web version driver

**mLink for Windows**  
 Win7 or Win10 (64-bit recommended) [Download](#)

**mLink for Mac**  
 macOS 10.12+ [Download](#)

**linux.deb** [Download](#)

**linux.rpm** [Download](#)

**mLink for Chromebook**  
 Supports using on Chromebook [Download](#)

**mBlock 3** (Stop updating) [Previous version >>](#)

**mBlock 3 for Windows**  
 Win7 + [Download](#)

**Win XP** [Download](#)

**mBlock 3 for Mac**  
 No longer work on macOS Catalina 10.15 and above [Download](#)

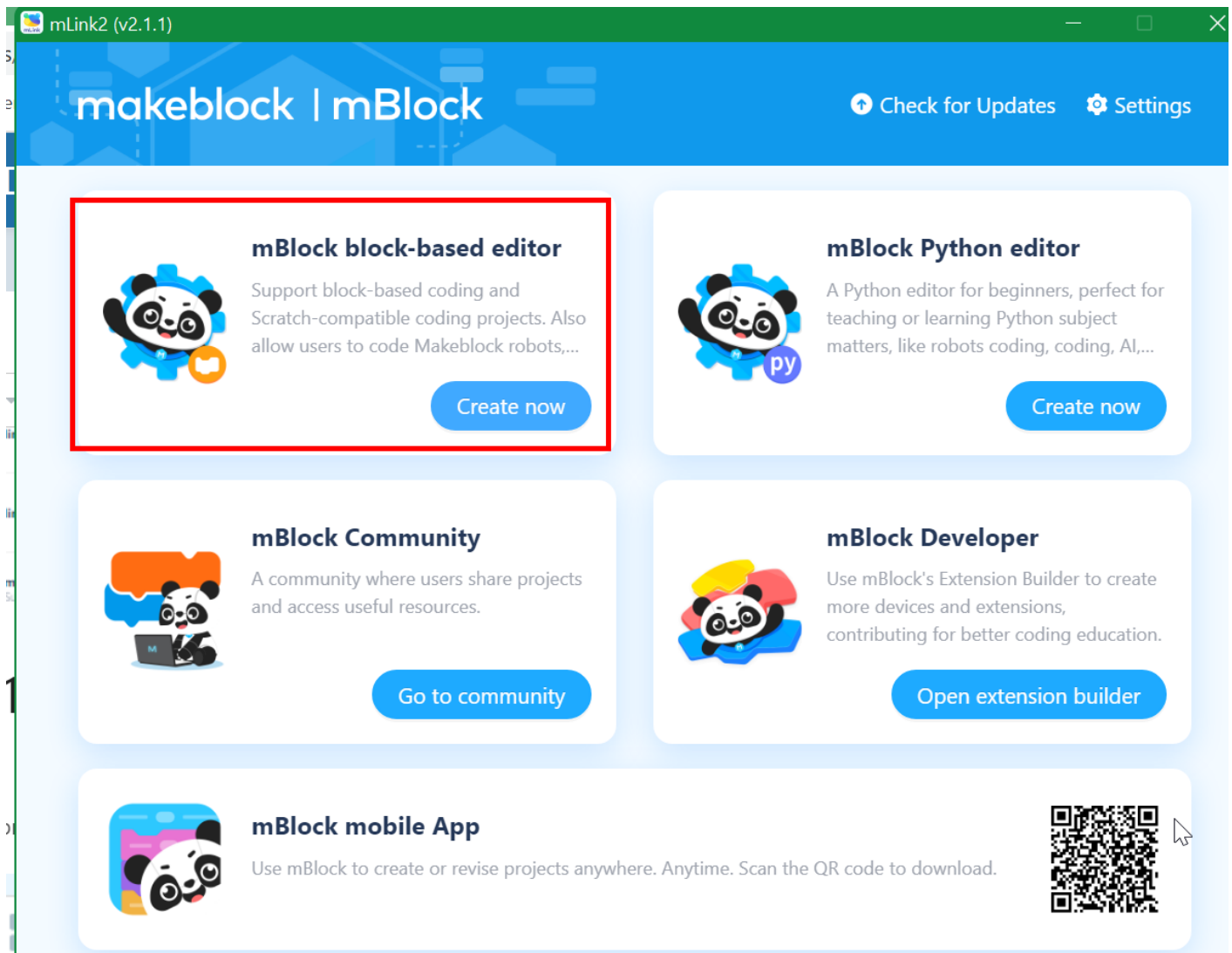
**mBlock 3 for Chromebook** [Download](#)

**mBlock 3 web version**  
 Will no longer be available from December 31, 2020. [why? >>](#) [mBlock 3 IDE](#)

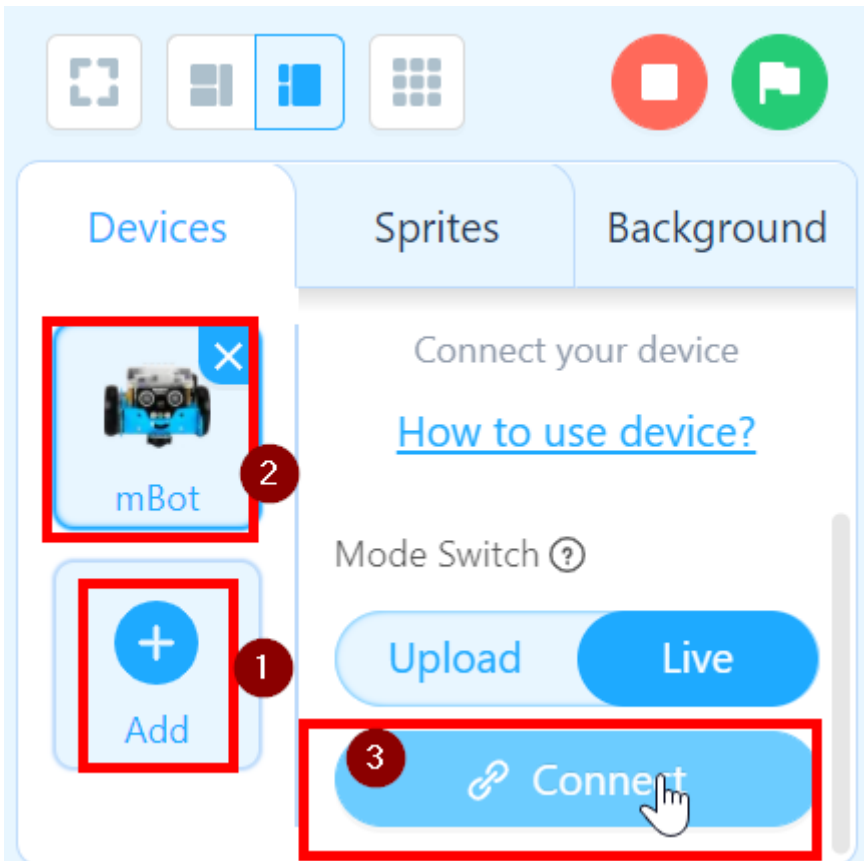


## 4.1.1 Versión Web

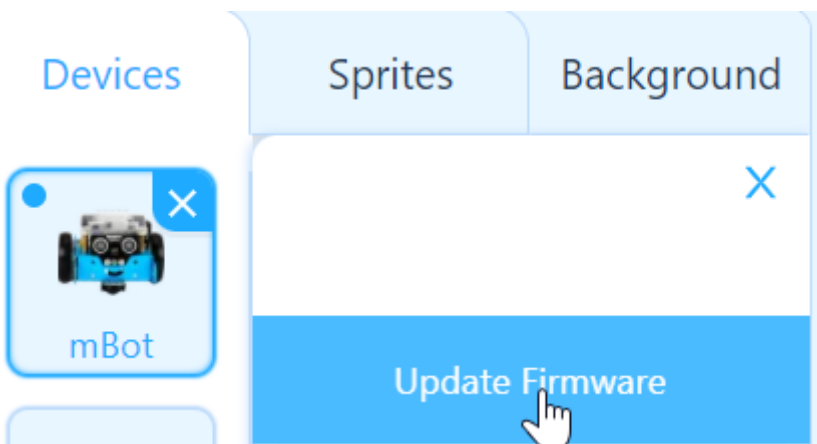
Después de lanzar **mLink** se abre las diferentes opciones de trabajar, elegimos programar en bloques



Se abre la página web <https://ide.mblock.cc/> en el cual tenemos que añadir el dispositivo mBot, y darle a conectar.



Si trabajamos **en vivo o live** pide instalar el firmware



En modo **cargar o upload** solo permite el evento *Cuando mBot se pone en marcha*. Una vez acabado el programa, le damos a subir:

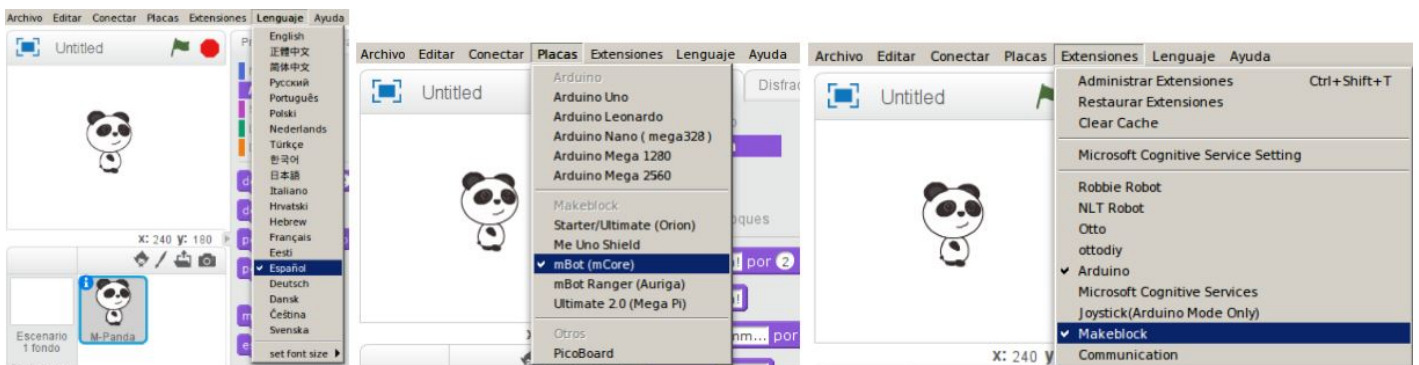


### 4.1.2 mBlock PC versión 5

En este caso descargamos el programa y no es necesario ejecutar **mLink**. La interfaz y el funcionamiento es exactamente igual que la versión web.

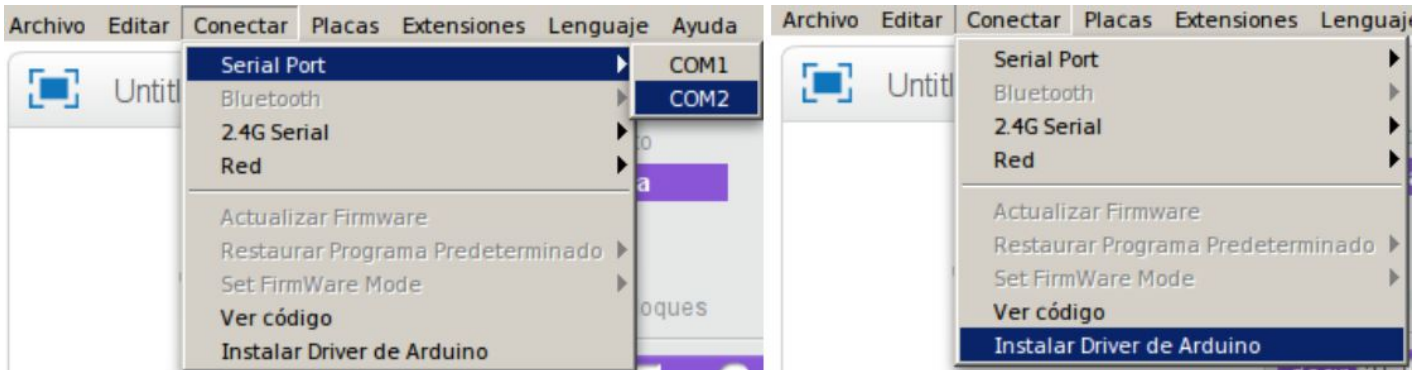
### 4.1.3 mBlock versión 3

Una vez dentro del programa, seleccionamos mBot



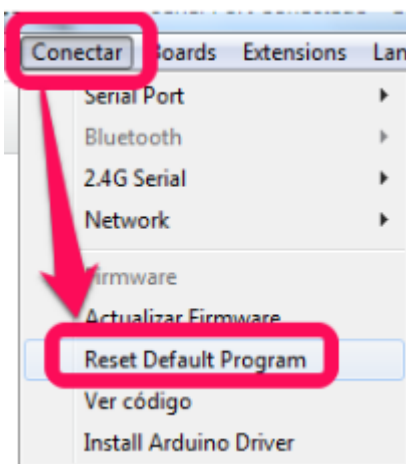
Fuente: <https://mclon.org> Maria L CC-BY-SA

Y lo conectamos por el puerto serie, teniendo nuestro mClon conectado por cable al ordenador. Si no lo detecta, instalar el driver de Arduino.



Fuente: <https://mclon.org> Maria L CC-BY-SA

Hay dos modos de funcionar: **Dependiendo del ordenador (o en vivo en versiones 5x o Web)**, tiene la desventaja de que va lento, pues el programa lo ejecuta el ordenador y hay que tener mClon conectado pero tiene la ventaja que podemos interactuar con el ordenador. Para ello hay que instalar el programa por defecto:



Fuente: <https://mclon.org> Maria L CC-BY-SA

O el modo **independiente del ordenador (modo Cargar en versiones 5x o Web)** donde el programa se ejecuta en el mismo mClon y puede ir desconectado del ordenador, es el método más aconsejable al menos que se quiera interactuar con el ordenador.



```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 #include <MeMCore.h>
6
7 MeDCMotor motor_9(9);
8 MeDCMotor motor_10(10);
```

Fuente: <https://mclon.org> Maria L CC-BY-SA

Si quieres más información de la forma de programar con mBloc, visita el curso mBot en [www.catedu.es](http://www.catedu.es) en tutoriales

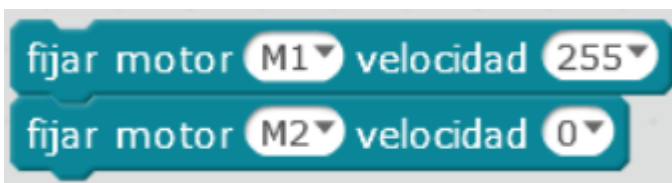
4 Testeo

## 4.2 Testeo con mBlock motores

Ahora hay que hacer programas sencillos de los diferentes elementos. La forma más rápida y sencilla es **el modo live, en vivo o dependiendo del ordenador** (como quieras llamarlo).

Si alguno no funciona, hay que comprobar las conexiones, verificar continuidad con un polímetro... etc.

Ponemos el mClon **dependiendo del ordenador** y hacemos dos clics en esta instrucción :



Si no gira el motor, a lo mejor es que están intercambiados el izquierdo y el derecho, ejecuta pues este programa.



Si tampoco funciona, revisar el pin VM que tiene que estar conectado a +5V Ard.

Si el giro de algún motor no es el apropiado, es que están intercambiado su polaridad.

Si uno de los dos nunca gira, revisa las conexiones entre controlador e inversor.

Si utilizas las versiones Web o 5x de mBlock cambia algo la instrucción



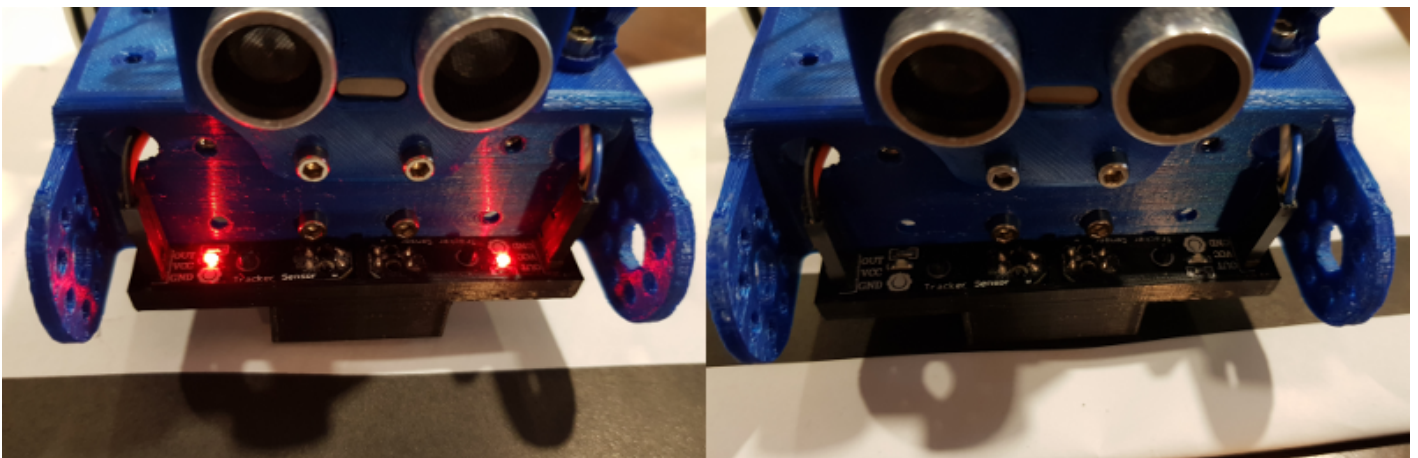
The screenshot displays the mBlock software interface. On the left, a sidebar contains a 'Devices' panel with an 'Add' button (1) and a connected 'mBlock' device (2). Below it are 'Upload' and 'Live' buttons (3) and a 'Disconnect' button (4). The central 'Action' category (5) lists several motion blocks: 'move forward at power 50 % for 1 sec', 'move backward at power 50 % for 1 sec', 'turn left at power 50 % for 1 sec', 'turn right at power 50 % for 1 sec', 'move forward at power 50 %', 'left wheel turns at power 50 %, right wheel turns at power 50 %' (6), and 'stop moving'. A 'left wheel turns at power 0 %, right wheel at power 50 %' block is also visible. The right side shows a workspace with a grid and a single block placed on it (7).

4 Testeo

## 4.3 Testeo con mBlock sensores

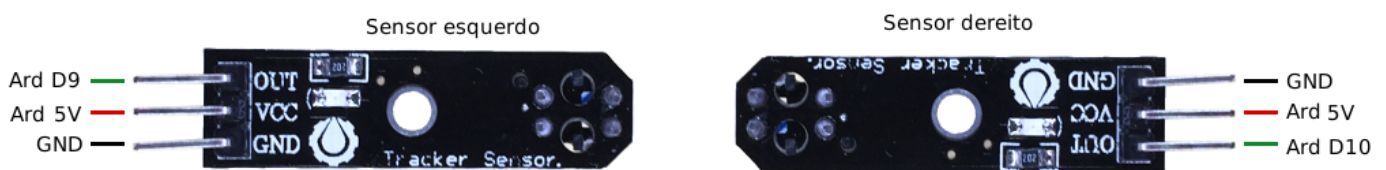
### 4.3.1 Test rápido del siguelíneas

Primera comprobación para el sigue líneas, primero hay que testear que funcionan correctamente su detección negro blanco, para ello utilizamos un folio con una línea negra y vemos si se encienden los leds cuando hay blanco y no se encienden cuando hay línea negra:



Fuente: <https://mclon.org> Maria L CC-BY-SA

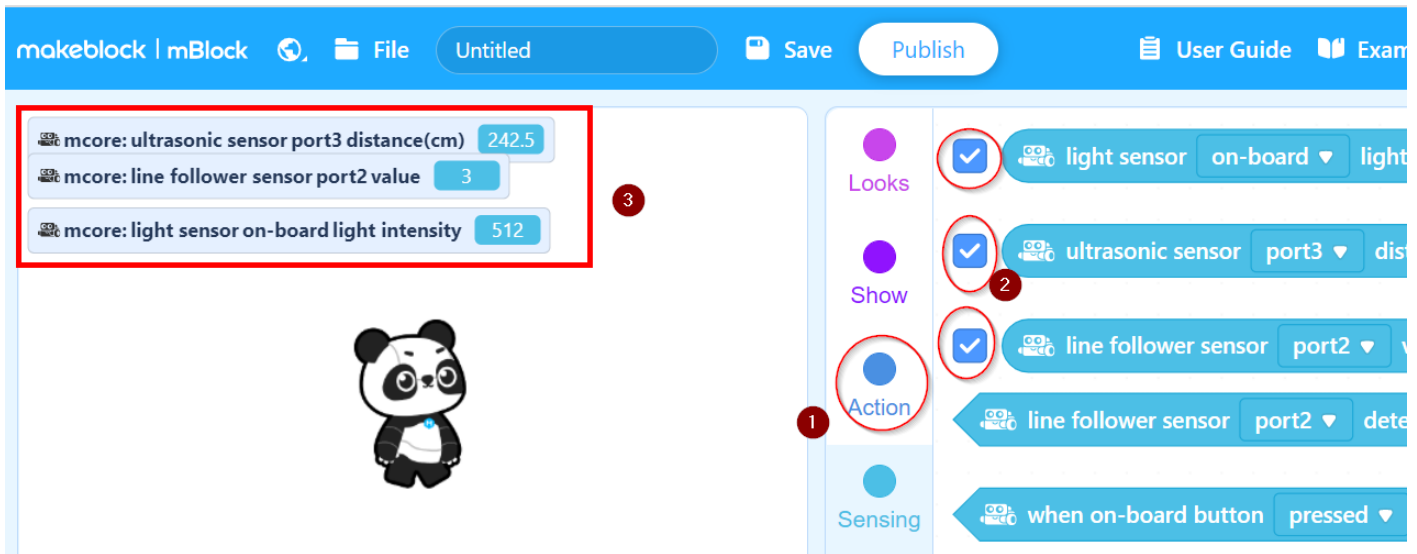
Si no funciona bien, es que los pines están mal conectados, es fácil confundirse pues va uno al revés del otro :



Fuente: <https://mclon.org> Maria L CC-BY-SA

### 4.3.2 Test con mBlock versión Web o ver 5x de los valores de los sensores

En modo **vivo** es tan fácil como hacer *check* en los diferentes sensores. Arriba en el panda enseña sus valores



En el **sigue líneas** tiene que salir los siguientes valores :

- **0** si detecta todo negro (en binario 00)
- **1** si detecta blanco derecha pero negro a la izquierda (01)
- **2** al revés (10)
- **3** si detecta los dos blancos (11)

Si funciona al revés es que se han intercambiado los pines D9 y D10

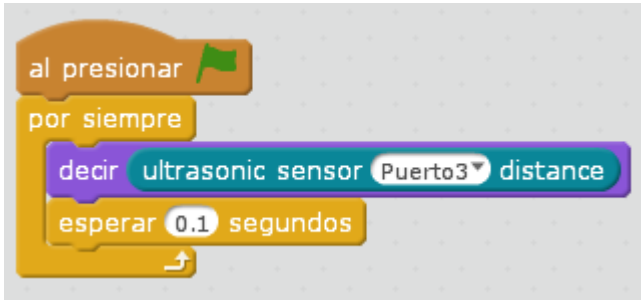
En el **sensor de ultrasonidos** la distancia en cm del obstáculo enfrente de los "ojos"

En el **sensor de luz**, el valor ambiental, es interesante tomar nota de este valor para crear programas en el futuro, ver cual es el **umbral de la oscuridad**, es decir, qué valor es cuando tenemos la habitación o aula con luz, y cual es el valor con el mismo lugar a oscuras. Por ejemplo ver estos programas del [Scrath Day Nerja](#)

## 4.2.3 Testeo con mBlock ver 3x de los valores de los sensores

### SENSOR ULTRASONIDOS

Un programa sencillo es el siguiente en el modo **dependiendo del ordenador**



Fuente: <https://mclon.org> Maria L CC-BY-SA

Y el oso panda nos va diciendo las distancias:



## SENSORES SIGUE LINEAS

Luego hacemos el siguiente test con mBlock **dependiendo del ordenador** y el osito panda nos dirá qué ocurre:

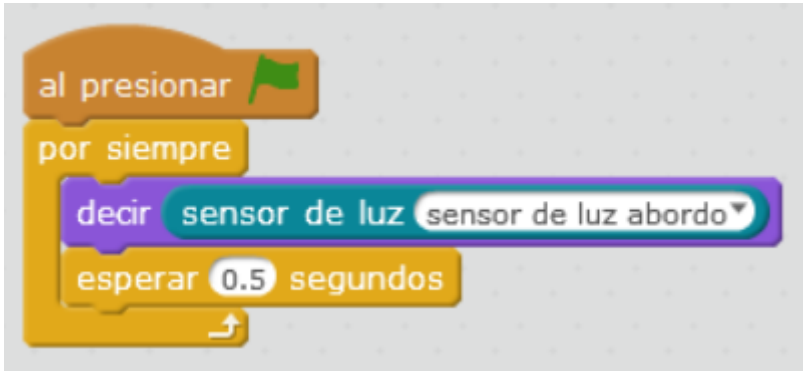


Fuente: <https://mclon.org> Maria L CC-BY-SA

Si funciona al revés es que se han intercambiado los pines D9 y D10

## SENSOR DE LUZ

Recomendamos usar este programa en el modo **dependiendo del ordenador o en vivo**



El muñeco panda nos irá diciendo la cantidad de luz

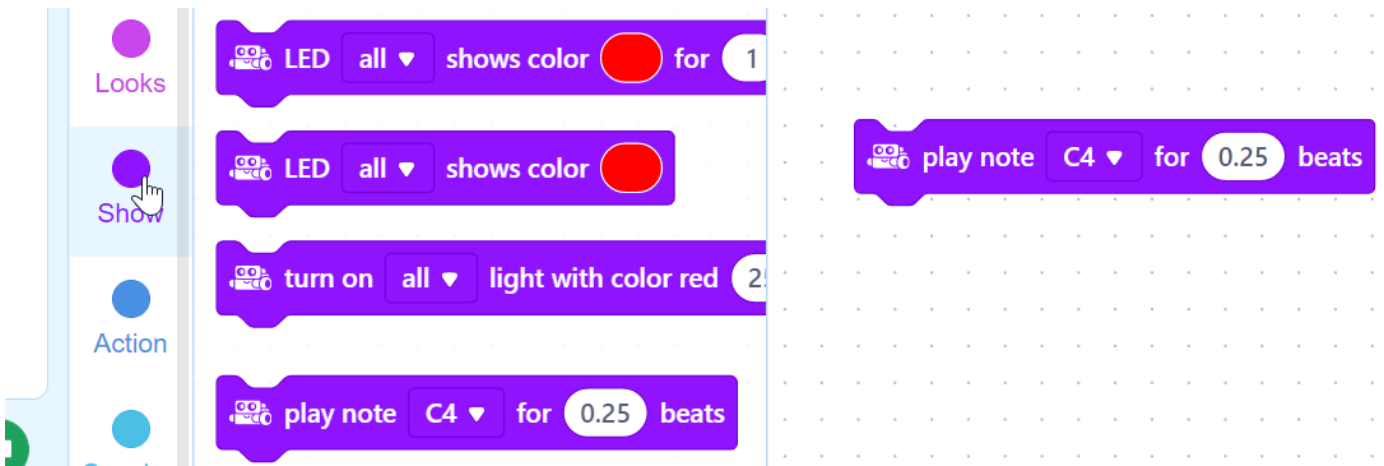
Este programa nos servirá para definir el **umbral de oscuridad** descrito anteriormente

4 Testeo

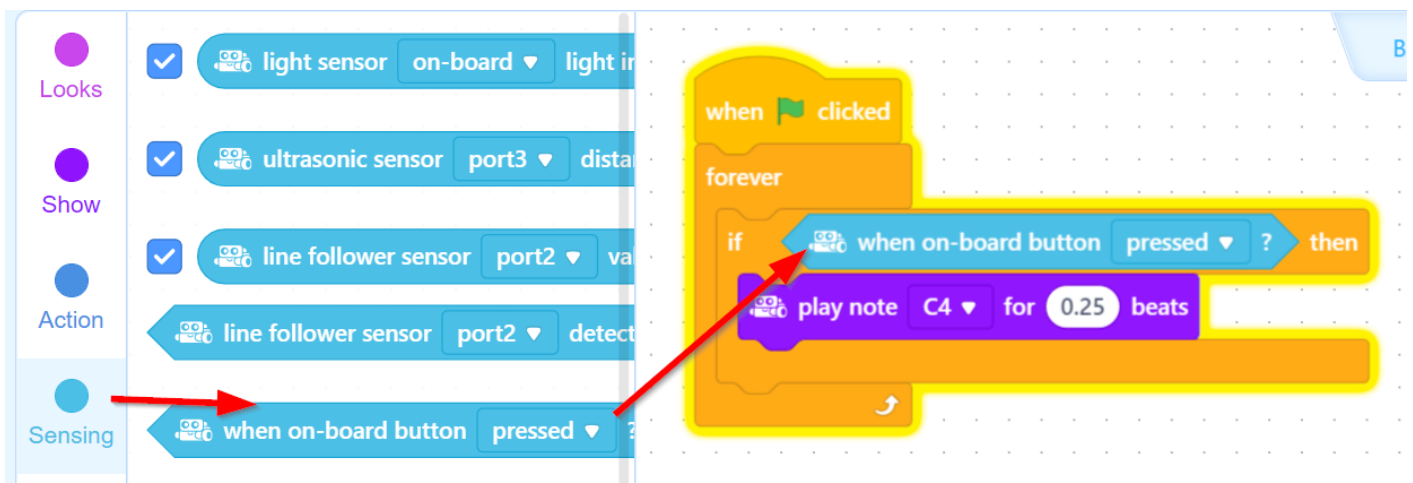
# 4.4 Testeo con mBlock accesorios

## 4.4.1 Con mBlock versión Web o 5x

El zumbador es muy fácil comprobarlo con la conexión **live o en vivo** de nuestro mClon



Una vez que ya está comprobado, realizamos este programa para testear el pulsador



## 4.4.2 Con mBlock versión 3x



## ZUMBADOR

Este programa es sencillo en el modo **independiente del ordenador**



Fuente: <https://mclon.org> Maria L CC-BY-SA

## PULSADOR

Aprovechando que el zumbador ya funciona, hacemos este programa de forma **independiente del ordenador**:



Fuente: <https://mclon.org> Maria L CC-BY-SA

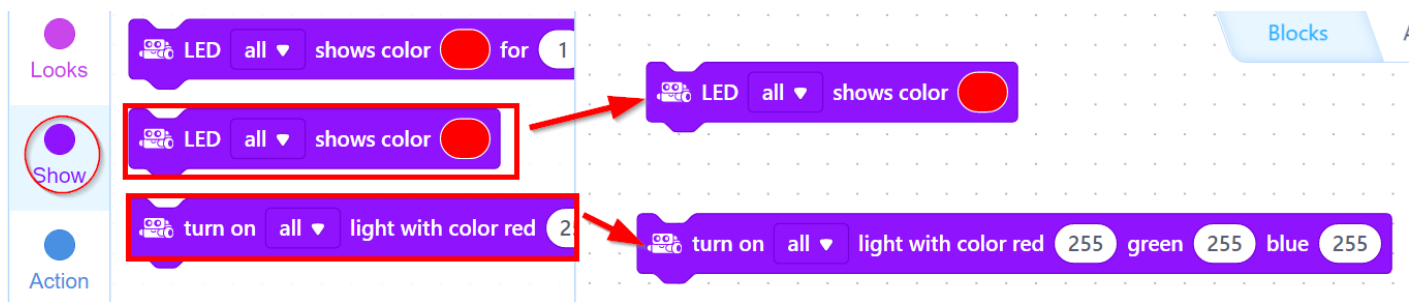
4 Testeo

## 4.5 Testeo con mBlock extras

### 4.5.1 Con mBlock versión Web o 5x

#### LEDS

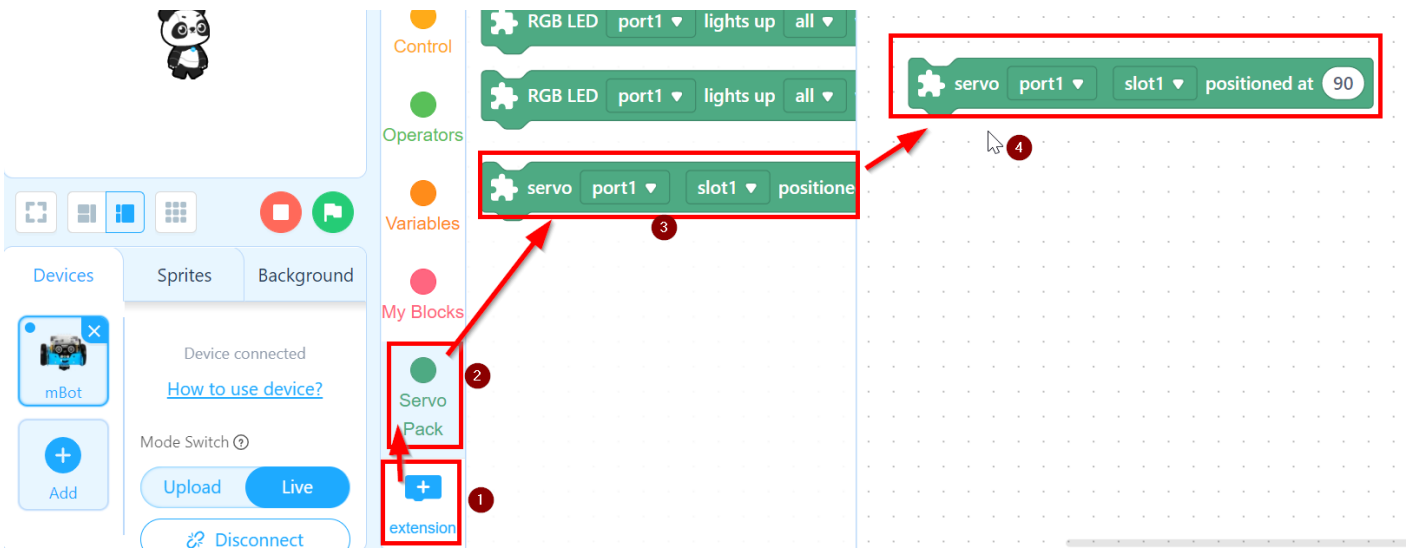
En modo **live o vivo** tenemos dos instrucciones, al hacer *clik* en el primero podemos elegir el color, en el segundo tenemos más control de los valores, por ejemplo podemos hacer el blanco total 255,255,255 que la primera instrucción no puede:



ya sabes que hemos explicado que el rojo y el azul están intercambiados

#### BRAZO

No está por defecto la instrucción, tenemos que instalarla en **Extensión - Paquete servo** y en modo **live o vivo** podemos ejecutarlo con dos cliks



Es el momento adecuado para **calibrarlo**, desmontar el brazo y colocarlo de forma que

- El ángulo de  $0^\circ$  corresponde a posición horizontal
- El ángulo de  $90^\circ$  corresponde a posición vertical

Una vez identificado cual es el ángulo  $90^\circ$  ejecutando la instrucción anterior, y una vez puesto el servo en esa posición, poner el brazo levantado con un ligero ángulo hacia delante, tal y como indica la figura :



Fuente: Adaptado de <https://mclon.org> Maria L CC-BY-SA

Si va muy deprisa, puedes ralentizarlo con el programa que se describe más abajo.

## 4.5.2 Con mBlock versión 3x

### BRAZO

Con el mClon **dependiendo del ordenador** hacemos dos clicks en esta instrucción, experimentando diferentes ángulos:

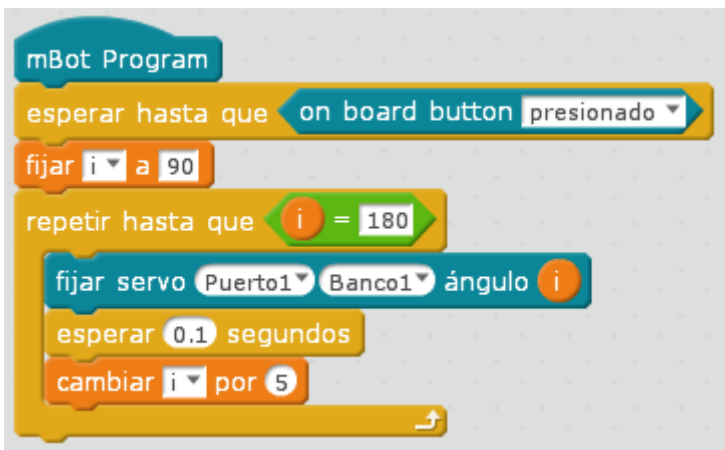


Una vez realizado, podemos experimentar este programa **independiente del ordenador**



Fuente: <https://mclon.org> Maria L CC-BY-SA

Como podrás observar va muy deprisa, una solución es poner en ángulo una variable que va aumentando:



Fuente: <https://mclon.org> Maria L CC-BY-SA

## LEDS RGB

Pon mClon **dependiendo del ordenador** y ejecuta dos clicks en esta instrucción:



ya sabes que hemos explicado que el rojo y el azul están intercambiados

#### 4 Testeo

## 4.5 Testeo completo con mBlock

Para superar el curso, tienes que entregar un vídeo donde se demuestre que funcionan todos los elementos :

- Motores
- Sensores
  - Sensor ultrasónico
  - Sensores sigue-líneas
  - Sensor luz
- Accesorios
  - Botón
  - Buzzer
- Extras
  - Leds RGB
  - Brazo

### 4.5.1 Propuesta TEST-CATEDU

Puedes hacer tú mismo un programa, una propuesta **que englobe los 8 elementos expuestos**, o si quieres, te proponemos esta propuesta que te la puedes descargar aquí [TEST-CATEDU.sb2](#)

La plantilla del circuito la podeis descargar [aquí imprimiendo 4 hojas del fichero Word](#) pero puede ser otra cualquiera.

<https://www.youtube.com/embed/6lOyVTUxPSo>

Tiene varios bloques definidos, este bloque **SIGUE LINEA** cumple la función de seguir la línea negra, pero si detecta un 3, llama al bloque **Buscalinea** pues está fuera del circuito:



El bloque **BuscaLinea** elige un número al azar y el 80% de veces retrocede a buscarlo, el otro 20% busca girando. Esto se hace así porque algunas veces se queda "atascado" hacia delante y hacia atrás, ese 20% es para romper ese ciclo vicioso.

```

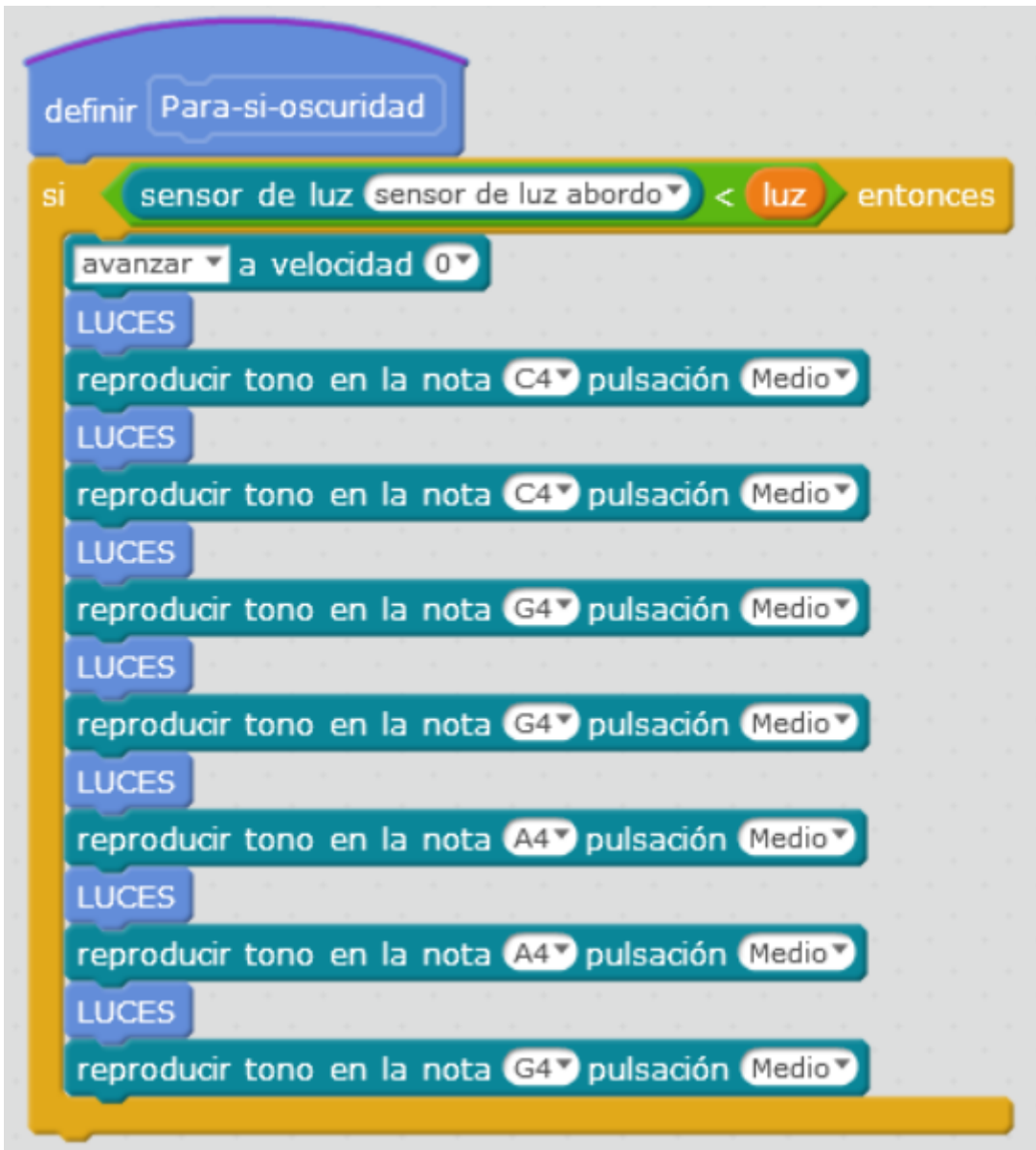
definiir BuscaLinea
fijar azar a número al azar entre 0 y 20
si azar = 0 entonces
  fijar motor M1 velocidad 0 - velocidad
si azar = 1 entonces
  fijar motor M2 velocidad 0 - velocidad
si azar = 2 entonces
  fijar motor M1 velocidad velocidad
si azar = 3 entonces
  fijar motor M2 velocidad velocidad
si 3 < azar entonces
  retroceder a velocidad velocidad
esperar 0.1 segundos
retroceder a velocidad 50
  
```

La función **VERLATA** mira si hay una lata delante, y en ese caso la retira :



Para probar el sensor de luces, hemos pensado [en esta idea](#) :

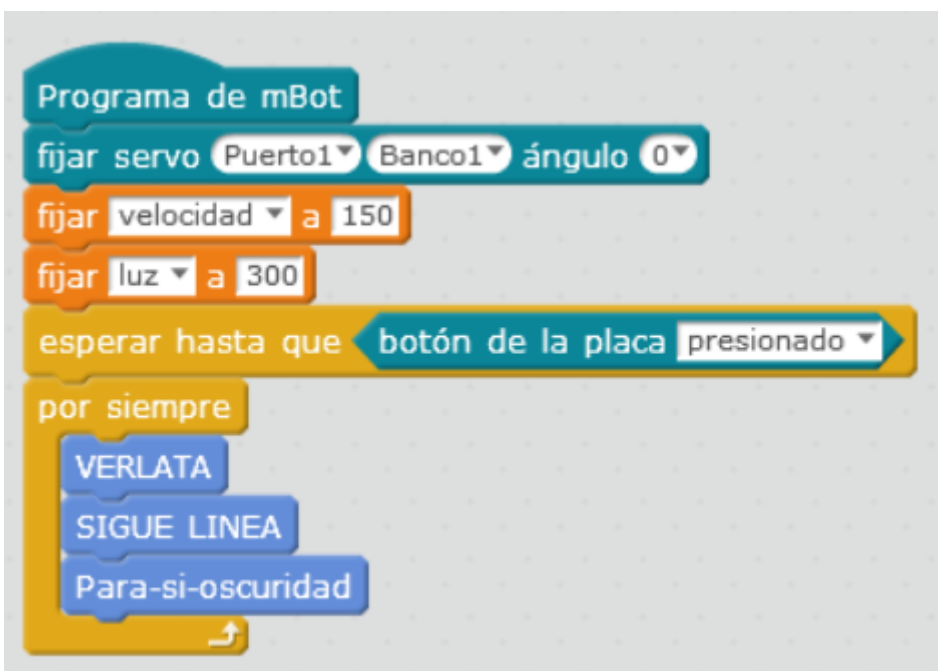
La función **Para-si-oscuridad** se detiene en el caso de que se apaguen las luces y reproduce un juego de luces con RGB y sonidos con el Buzzer:



Donde el bloque **LUCES** enciende los RGB al azar



Finalmente el programa principal que llama a todos los bloques es



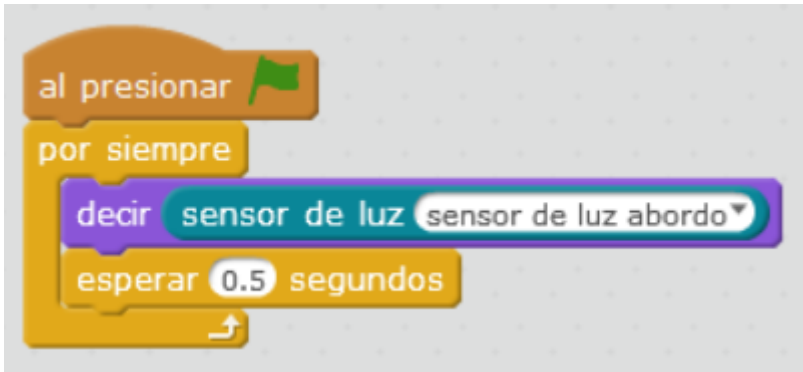
Donde tenemos que definir:

- La variable **velocidad** si es muy alta puede saltarse el circuito, si es muy baja puede que los motores no tengan suficiente fuerza para arrancar, depende del estado del PowerBank



- La variable **luz** que depende de las condiciones de la habitación, hay que elegir un valor lo suficientemente alto para que la luz detectada por el LDR sea mayor cuando las luces de la habitación están encendidas, y lo suficientemente baja para que cuando se apaguen, el valor medido por el LDR este más bajo de ese valor.

Para la variable **luz** recomendamos usar el programa visto en [4.3 Testeo sensores](#)



## 4.5.2 Otros testeos

En la página <https://mclon.org/> puedes encontrar interesantes propuestas en el apartado mBlock :

- Movimiento <https://tecnologia.org/mclon/o-robot-en-movimiento/>
- Cerramiento en línea <https://tecnologia.org/mclon/recinto-con-lina-2/>
- Sigue líneas <https://tecnologia.org/mclon/unha-lina-no-chan/>
- Detectar objetos <https://tecnologia.org/mclon/detectar-obxectos/>
- Luces RGB <https://tecnologia.org/mclon/extras/luces-de-cores/>
- Brazo robótico <https://tecnologia.org/mclon/extras/brazo/>
- Siguiendo una luz <https://tecnologia.org/mclon/extras/seguindo-a-luz/>

4 Testeo

## 4.6 Test con IDE

No hay que olvidar que mBot y mClon son en el fondo Arduinos, aunque sus placas no sean exclusivamente Arduinos UNOs. Por lo tanto podemos programar :

- De forma gráfica como hemos visto con mBlock y otros como ArduinoBlocks, Visualino ...
- De forma textual, con el programa **IDE de Arduino**.

Este programa se puede descargar en <https://www.arduino.cc/en/software>

No vamos a entrar, pues se escapa de las intenciones de este curso que es montar un mClon y probarlo, y esto ya se ha visto con programación por bloques con mBlock que es donde se obtiene todo el potencial de mClon al ser compatible con mBot.

No obstante, si quieres programarlo con texto, te recomendamos visitar la página

<https://mclon.org>:

## Propuestas con código :

- Como conectar el IDE a mClon <https://tecnologia.org/mclon/o-ide-de-arduino/>
- Pruebas test con IDE <https://tecnologia.org/mclon/probas-de-funcionamento-ide/>
- Control motores <https://tecnologia.org/mclon/o-control-dos-motores/>
- Cerramiento con línea <https://tecnologia.org/mclon/recinto-con-lina/>
- Sigue líneas <https://tecnologia.org/mclon/seguir-lina-basico/>
- No chocar. Sensor ultrasónico <https://tecnologia.org/mclon/ultrasonico/>
- Zumbador <https://tecnologia.org/mclon/o-zumbador/>
- Luces RGB <https://tecnologia.org/mclon/extras-ide/luces-de-cores-ide/>
- Brazo robótico <https://tecnologia.org/mclon/extras-ide/brazo-ide/>

Tamén esquivo obstáculos! Olo cos sensores ultrasónicos chineses.  
Dependendo do fabricante algúns van perfectos e outros saen algo rariños.

[pic.twitter.com/z7p1cl9VUb](https://pic.twitter.com/z7p1cl9VUb)

— mClon (@mClonRobot) [November 9, 2018](#)

# Muro

En este **Padlet** <https://padlet.com/CATEDU/mclon> puedes poner tus ejemplos o ejemplos que veas interesante para mostrar a los demás

<https://padlet.com/embed/fvinoj8ocb6c>

Hecho con Padlet

# Grupo telegram mClon

Para **mClon** hay un grupo de Telegram :

[mClonRobot](#)

mClon

162 members

Pinned message

Documentación disponible en [mclon.org](http://mclon.org)



Y un canal de Twitter @mClonRobot

# Créditos

Adaptado de <https://mclon.org/> por Javier Quintana CATEDU.

Autores de <https://mclon.org/>:

- **María Loureiro.** Catedrática de Tecnología en IES Primeiro de Marzo, Baiona. @tecnoloxia
- **Bernardo Álvarez.** Catedrático de Tecnología en IES de Teis, Vigo. @biober
- **Miguel Gesteiro.** Maker, Vigo. mgesteiro

Licencia: CC-BY-SA



**GOBIERNO  
DE ARAGON**

Departamento de Educación,  
Cultura y Deporte

**CATEDU**   
CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN