
TUTORIAL

ComPacto, sistema de educación robótico

ACERCA DE ROBÓTICA ALTERNATIVA



Robótica Alternativa es un proyecto que indaga en una nueva mirada sobre la educación robótica, poniendo el énfasis en la idea de la autoproducción de las herramientas robóticas como un proceso pedagógico y emancipador de la cultura de consumo desmedido.

Este proyecto es en parte resultado de la investigación “Una alternativa robótica para la Educación”, proyecto de la convocatoria **Untref 2018-2019**.

UNTREF

UNIVERSIDAD NACIONAL
DE TRES DE FEBRERO

Robótica Alternativa surge de la experiencia transitada en espacios alternativos de educación en arte y tecnología, donde el foco está puesto en incentivar las capacidades creativas y diversas de cada uno de los estudiantes. Es por ello que abogamos por un proyecto donde se libere el desarrollo creativo personalizado, fomentando el conocimiento y la apropiación crítica de las herramientas tecnológicas. En contraposición a muchos productos robóticos del mercado que pretenden homogeneizar la imaginación con sistemas desarrollados en otras culturas, este proyecto indaga en la posibilidad del desarrollo de un diseño robótico con identidad regional. Solamente experimentando con nuestra propia cultura material y simbólica se lograra en un futuro alcanzar nuestros propios desarrollos tecnológicos.

Este proyecto plantea la posibilidad de desarrollar actividades para la construcción de robots con distintos niveles de dificultad, favoreciendo la inclusión socieducativa a un amplio rango de edades de estudiantes y con una diversidad de contenidos educativos. Puede experimentarse con el Hardware, con el Software y con la materialidad de su diseño. Estos tres pilares son abiertos, modificables, mejorables, personalizables y compartibles. Es una búsqueda estética en todos sus aspectos.

Con esta visión, desarrollamos un sistema robótico flexible y permite incluir a estudiantes de los todos los niveles (primario, secundario y universitario), estimulando el pensamiento computacional, la lógica, el manejo de la complejidad y el trabajo colaborativo. Además puede ser utilizado en las aulas para afianzar conocimientos de física, matemática y mecánica, como también de plástica, biología y contenidos vinculados a los campos tradicionales del saber. Es una herramienta creativa de aprendizaje transversal a los contenidos de la currícula. Un espacio de aprendizaje, juego, encuentro y colaboración dentro y fuera del aula.

ACERCA DE COMPACTO



Compacto
robótica.alternativa

Es un sistema de educación robótico aplicable a distintos niveles educativos. El proyecto se establece dentro del pensamiento **Open Design, Open Hardware y Open Software**. Propone la búsqueda de una identidad robótica. **ComPacto**, no es un Kit robótico, es un **Concepto** en educación Robótica. El foco de la actividad esta puesto en incentivar las capacidades creativas diversas de cada uno de los estudiantes. Utilizando piezas modulares y distintos tipos de descartes (tecnológicos, plásticos, maderas, etc.) cada equipo podrán construir un robot, dándole su carácter físico, su propia personalidad. Luego del diseño físico del robot, se define su comportamiento, a partir de la determinación de una secuencia de movimientos y acciones a través de la programación.

¿Cómo construir ComPacto?

PASO N° 1 MATERIALES

Para construir **ComPacto**, necesitarás conseguir los siguientes materiales (ver imágenes)

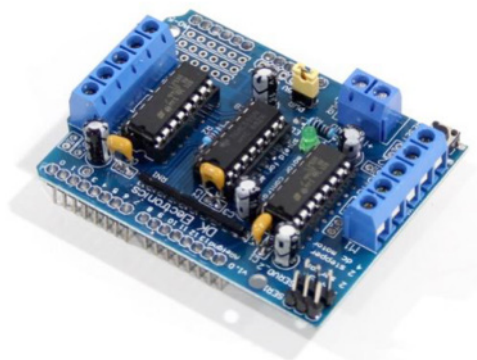
- 1 Arduino Uno
- 1 Shield para Arduino Puente H L293d
- 2 Motor con rueda (DC de 5 a 6 volt hasta 500 ma)
- 1 Sensor Ultrasónico hc-sr04 o similar.
- 2 sensores LDR (de 5 milímetros o similar).
- 2 sensores Infrarrojo CNY70 o similar.
- 2 Pilas Li-ion Recargables de 3.7 volts.
- 1 porta pila para 2 Pilas Li- ion.
- 1 Interruptor Palanca Llave
- 1 conector Dupont Terminal Hembra 2.54 Armar Cable Arduino x50.
- Una tira de pines 2 vías 0.1 in hembra.
- Una tira de pines simple hembra para cable 2.54 mm (0.1 in)
- Una tira de pines doble hembra para cable 2.54 (0.1 in)



Arduino UNO



Dos motores
con rueda



Shield para Arduino
Puente H L293d



Interruptor
palanca Llave



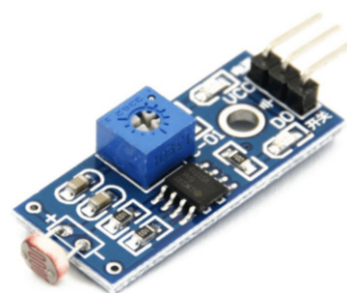
Pilas Li-ion
Recargables x 2u.



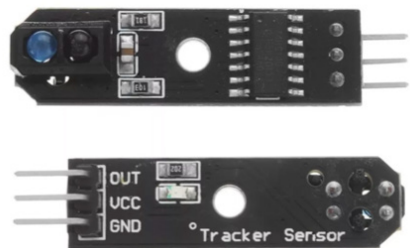
Un sensor
ultrasónico



Porta pila para 2
Pilas Li-ion



Dos sensores
LDR



Dos sensores
Infrarrojo



Una tira de pines
simple hembra
para cable 2.54
mm (0.1 in)



Conector Dupont
Terminal Hembra
2.54 x50u.



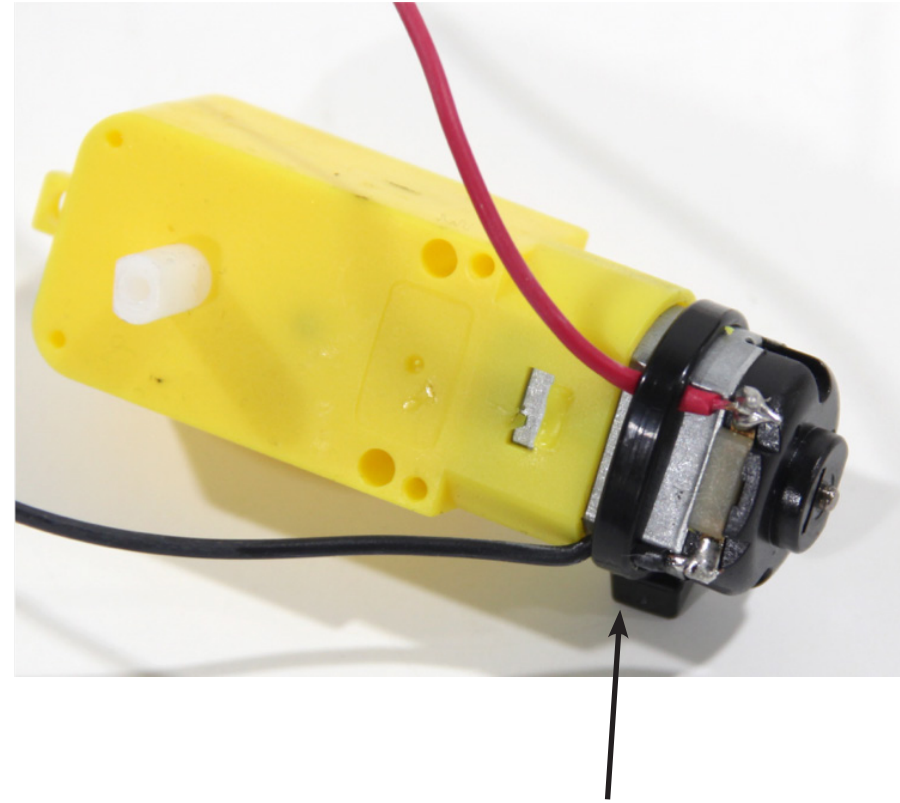
Una Tira de pines
doble hembra
para cable 2.54
(0.1 in)

PASO N° 2 MOTORES



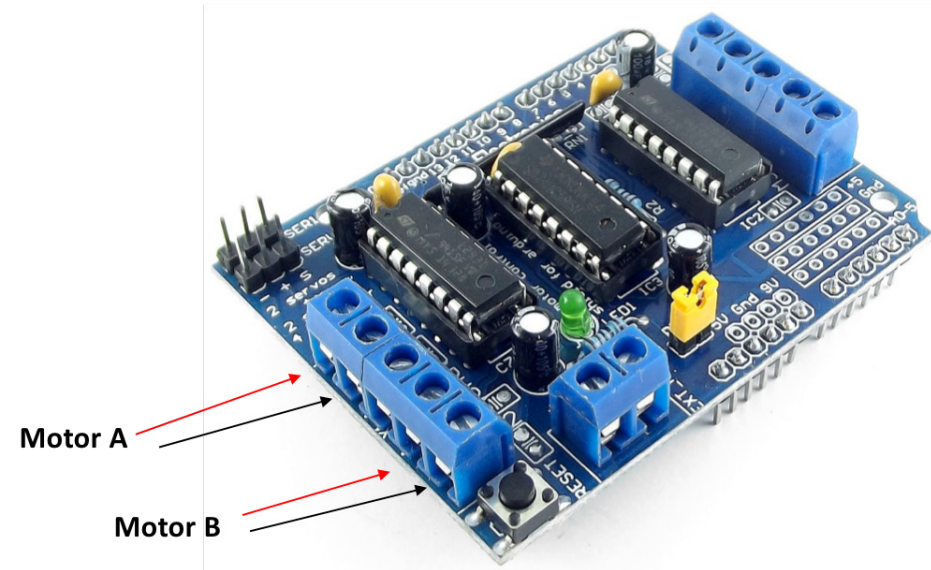
A continuación, vamos a soldar dos cables a las terminales del motor. Si es posible de distintos colores.

PASO N° 2.1 Seguridad de los cables.



Si es posible, sujetá los cables con un **precinto** pasado alrededor del motor. Esto es para evitar que se rompa los terminales de los motores si se tira de los cables.

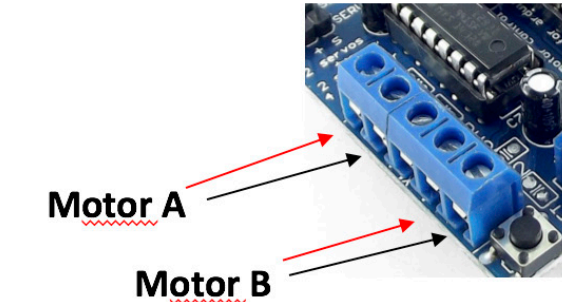
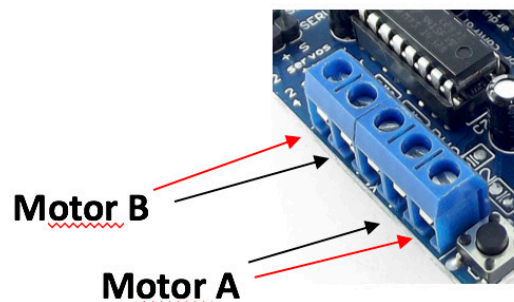
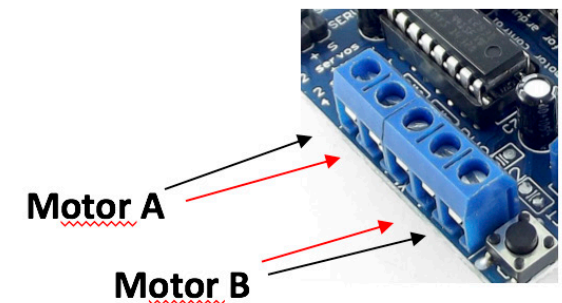
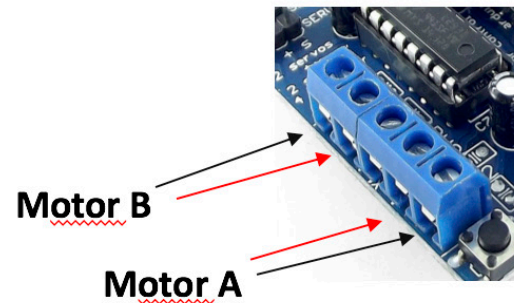
PASO Nº 2.2 Conexión



Los motores van conectados estas borneras del **Shiel L293**. Las puntas de los cables, sin vaina, se deben ajustar con un destornillador. La bornera central siempre queda libre, no se usara. El orden de los cables depende de como están soldados en el motor y como esté armado el robot, por ello no se puede especificar la forma de antemano, pero **ATENCIÓN: no hay que mezclar los cables de los motores**. A continuación un ejemplo:

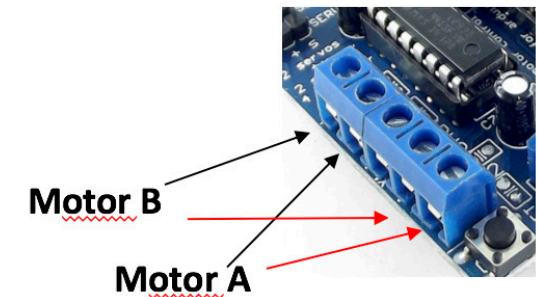
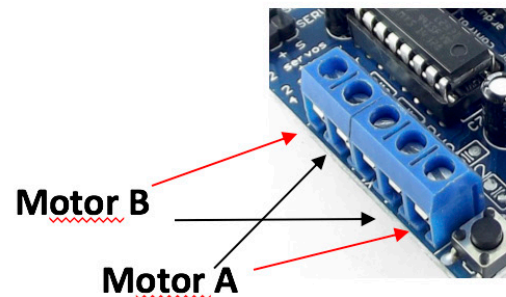
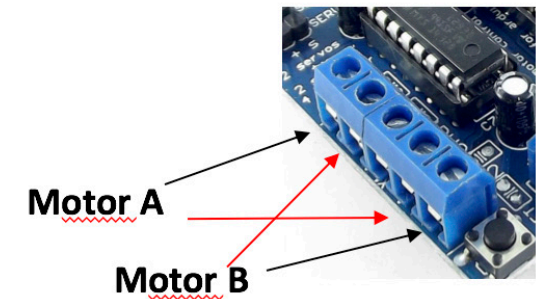
PASO N° 2.3 Conexión

La diferencia de conexión de una forma u otra es que el motor puede girar hacia el lado contrario de lo esperado. Si queremos que avance puede que retroceda. Por ello hay que probar las combinaciones posibles dependiendo de cómo esten conectados los cables de los motores. Todas las siguientes combinaciones son posibles:



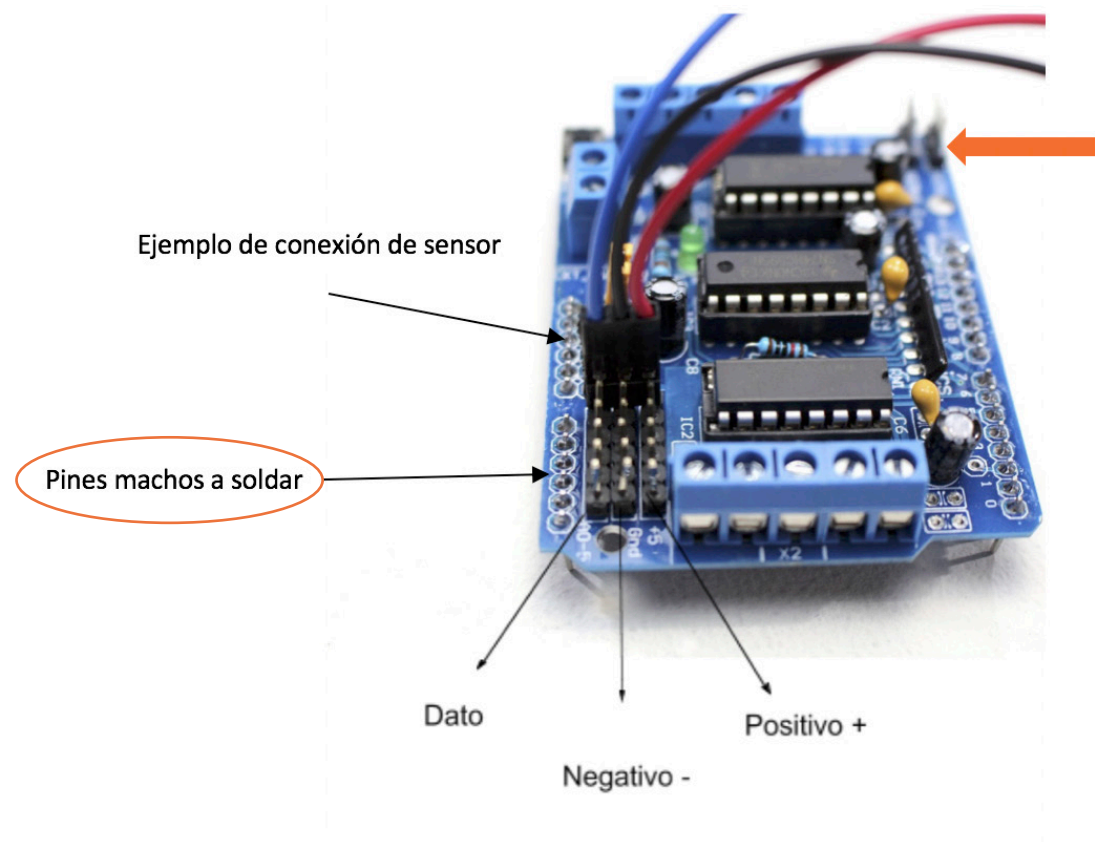
PASO N° 2.4 Conexión

Lo que no puede hacerse es mezclar los cables de los motores. **Los siguientes ejemplos NO deben hacerse.** Son unas de las tantas combinaciones que no deben realizarse.



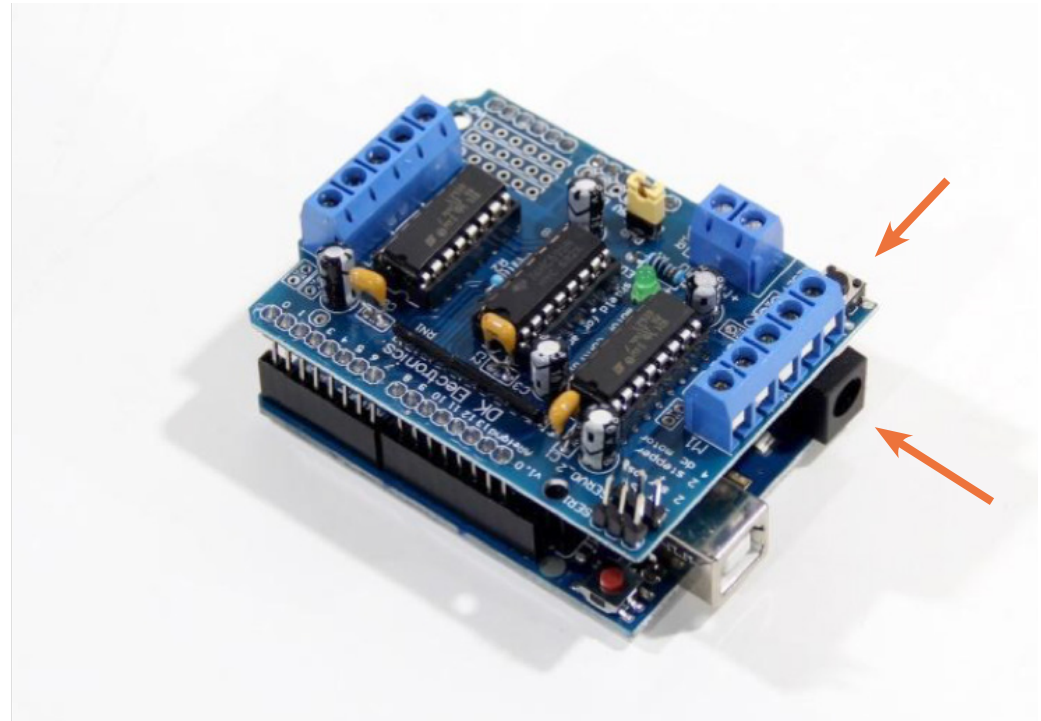
PASO N° 3

Preparación del Shield Puente H L293d



En este paso vamos a **soldar los pines machos** a la shield del Arduino para poder conectar los diversos sensores que usa **ComPacto**

PASO N° 4 ACOPLAMIENTO



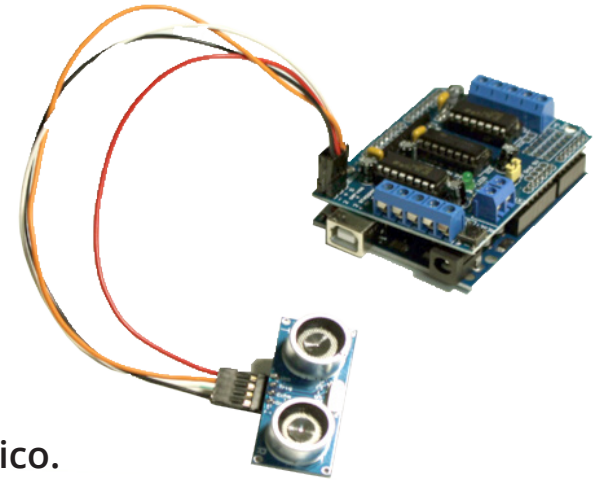
Para montar la shield sobre el Arduino, es necesario ubicar los pines en el lugar de encastre que corresponde.

Te puedes guiar por la ubicación del interruptor del shield y la ficha de conexión de voltaje del Arduino.

PASO N° 5

Conexiones del Sensor Ultrasónico

(necesario solamente para
la utilización de este sensor)



Conexiones del Sensor Ultrasónico.

Este sensor permite medir distancias, en centímetros, entre el robot y un obstáculo. En este sistema permite medir hasta un metro de distancia en una dirección recta al sensor. Mide dentro de un rango de 30 grados, es decir que el sensor es bástate dirección. Atención: el sensor reconoce obstáculos grandes (por ejemplo no reconoce patas de sillas, si una pared). Un típico error es que en sensor apunte al piso, colocarlo de forma que apunte hacia delante y levemente hacia arriba. Este paso es complejo por ello lo realizaremos de a pequeñas acciones.

PASO N° 5.1.1 Armado de cables.

Crimpeado de cables

En este paso, usaremos unas pinzas de punta plana para crimpear los cables al **Conector Dupont Terminal Hembra 2.54**. Debemos colocar el terminal el 4 cables de los dos lados.

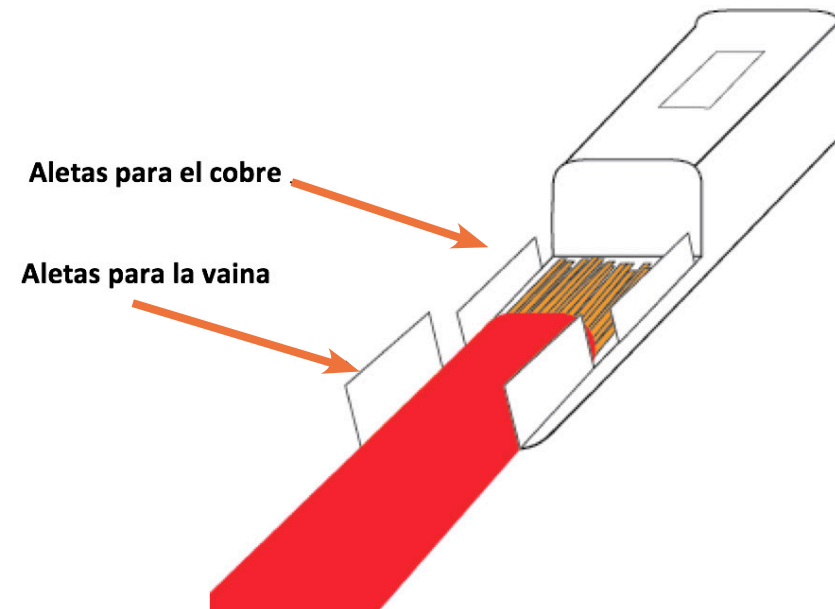


La medida de los cables debe ser de unos 25 cm aproximadamente y en preferencia de distintos colores para identificar fácilmente las conexiones.

PASO N° 5.1.2 Armado de cables.

Crimpeado de cables

La vaina del cable de estas sujeta por unas de las aletas. Y el cobre del cable debe estar apretado por las aletas internas haciendo contacto con el terminal.



PASO N° 5.1.3 Armado de cables.

Crimpeado de cables

El cable debe quedar apretado en el terminal firmemente. Y el cobre del metal en contacto con el terminal. Intentar que el cobre del cable no sobresalga por la parte trasera del terminal y que los hilos de cobre entren en el tubo del terminal.

Aletas para el cobre

Aletas para la vaina



PASO N° 5.2

Armado de conectores



Conexión de Cables

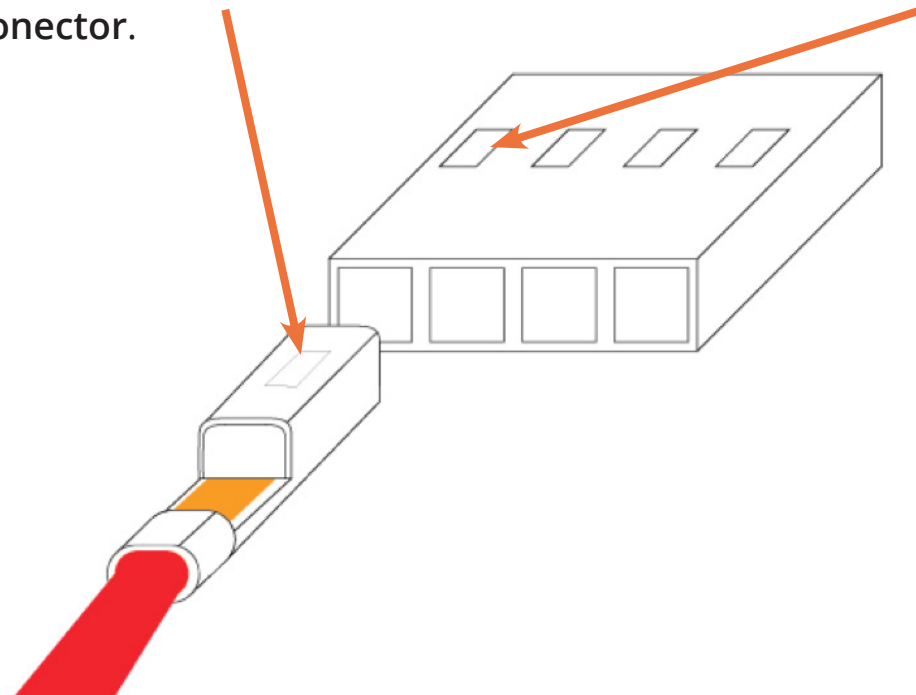
El próximo paso es colocar los cables con los terminales en el conector. Cada cable debe ir en una posición determinada.

ATENCIÓN: Cuidado!!!, si no se conecta adecuadamente es probable que se rompa el sensor o el Arduino.

PASO N° 5.2.1 Armado de conectores

Tira de pines simples.

Cortar de la tira de pines una pequeña ficha de **solo cuatro vías**. Y luego colocar un cable en cada vía del conector. La parte plana del terminal va hacia abajo. No se encuentra la parte plana con la perforación del conector. El terminal entra bien profundamente hasta que una **pequeña lengua del terminal** se encastra en la **perforación del conector**.

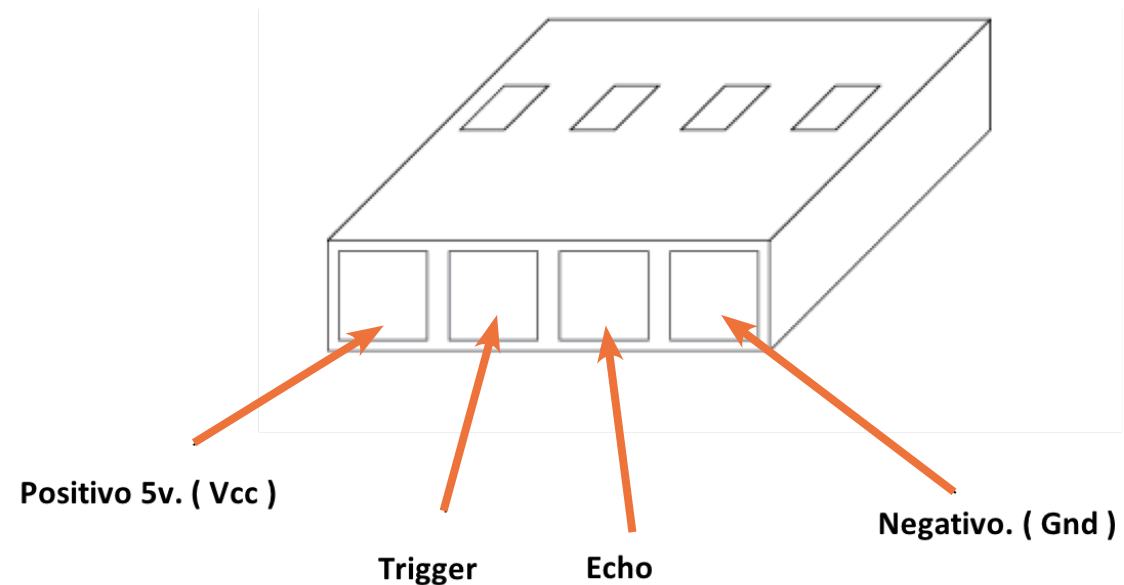


PASO N° 5.2.2

Armado de conectores

Tira de pines simples.

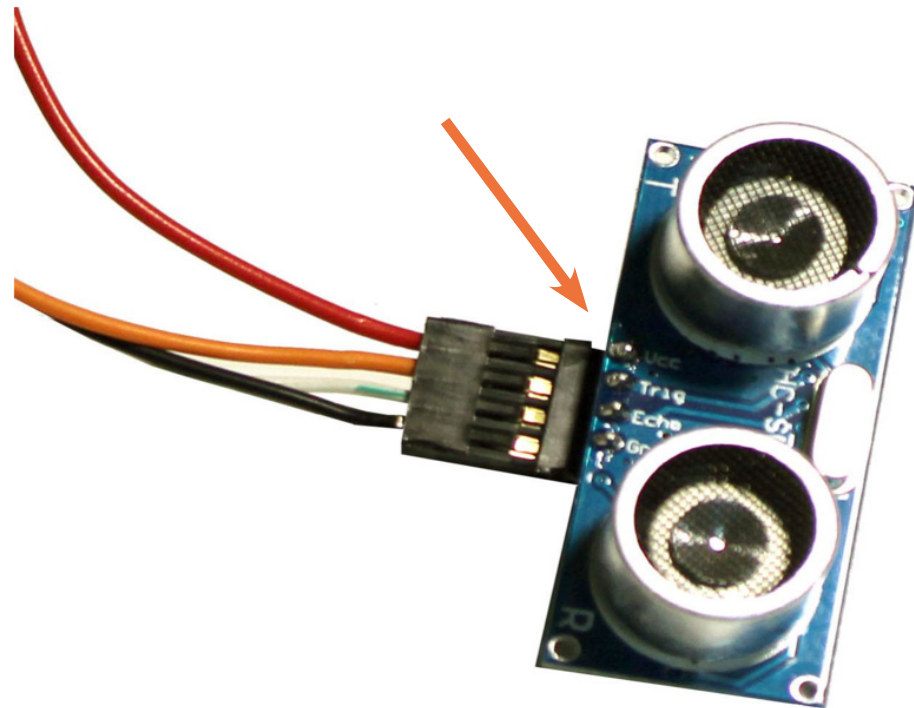
El orden de los cables para el sensor HC-sr04 es el que se muestra en la imagen. Lo que importa es recordar el color que se utilizó para luego hacer la conexión correcta del otro extremo del cable.



PASO N° 5.2.3 Armado de conectores

Conectar el cable al sensor ultrasónico.

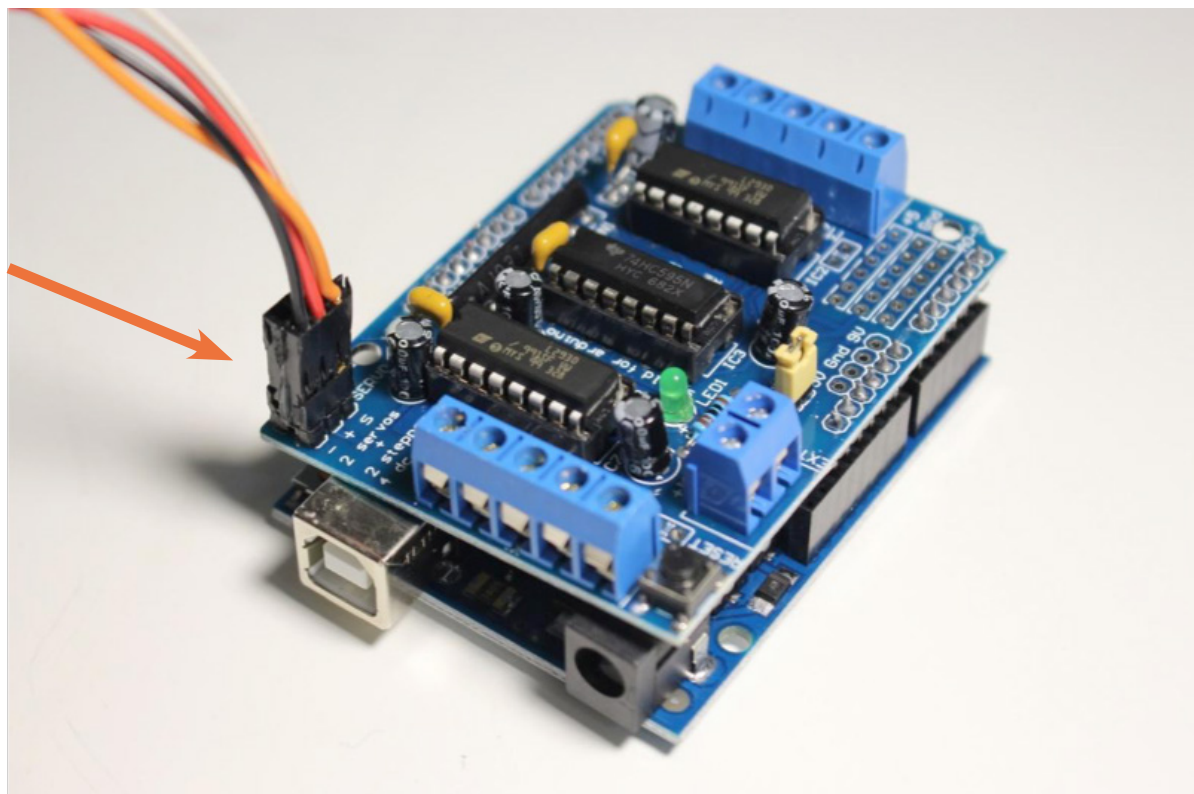
Cuidado!!. Poner el conector de forma que concuerden los cables designados con la inscripción que posee el sensor en cada pin.



PASO N° 5.2.4 Armado de conectores

Conectar los pines hembras dobles a la shield.

Ahora armaremos el conector del cable del sensor ultrasónico del lado del Shield.

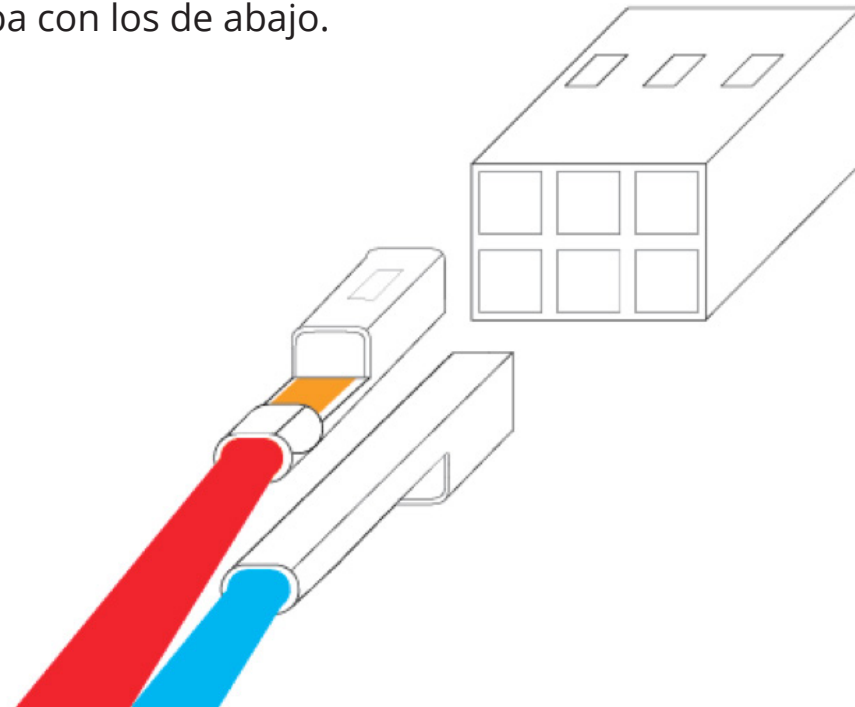


PASO N° 5.2.5

Armado de conectores

Tira de pines dobles.

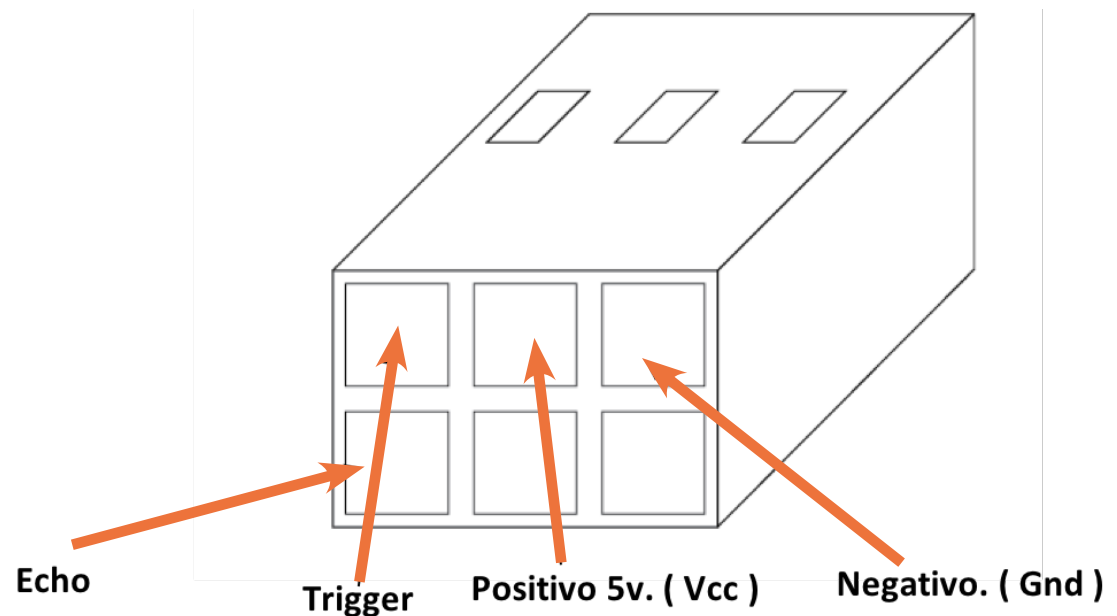
Cortar de la tira de pines dobles en una pequeña ficha de solo **seis vías**. Dos pisos de 3 vías cada uno. Luego colocar los cuatro cables en el conector. Tener en cuenta que los terminales de los cables van con la parte plana hacia el centro del conector. Por ello, van invertidos los de arriba con los de abajo.



PASO N° 5.2.6 Armado de conectores

Tira de pines dobles.

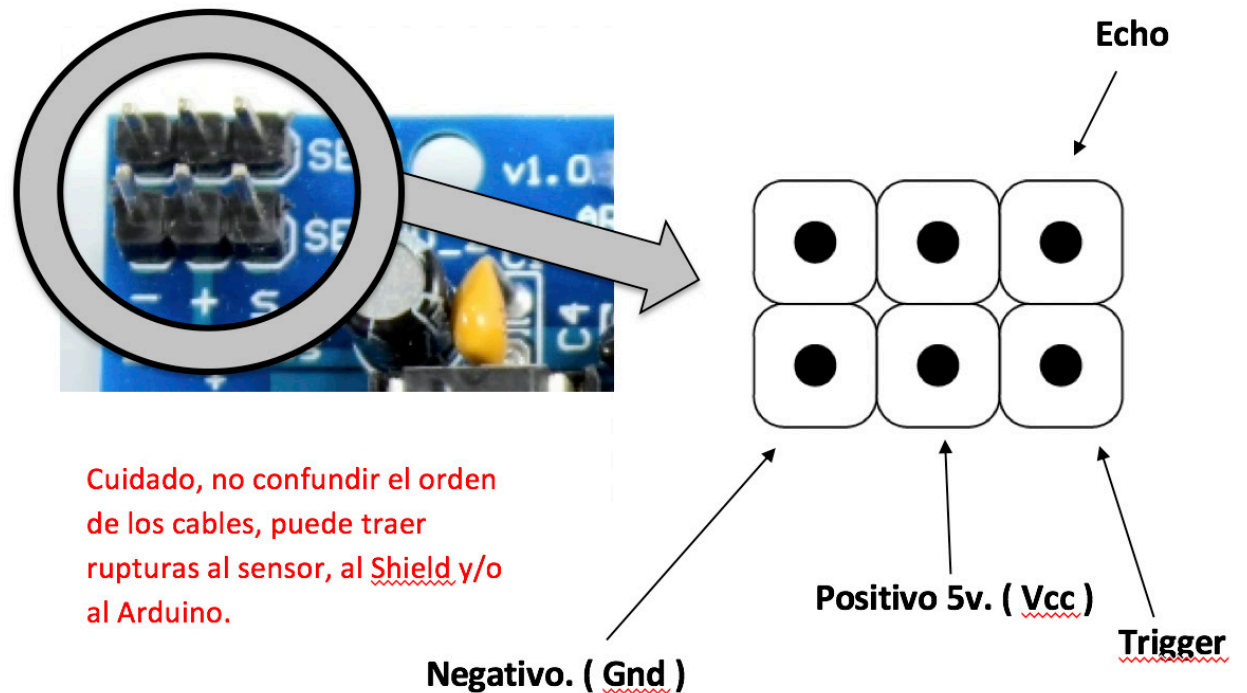
El orden de los cables para el sensor HC-sr04 del lado del Shield esta en el grafico siguiente. Cada cable tiene que respetar la conexión realizada el otro lado. **ATENCIÓN: Cuidado, no confundir el orden de los cables, puede traer rupturas al sensor, al Shield y/o al Arduino.**



PASO N° 5.2.7 Armado de conectores

Tira de pines dobles.

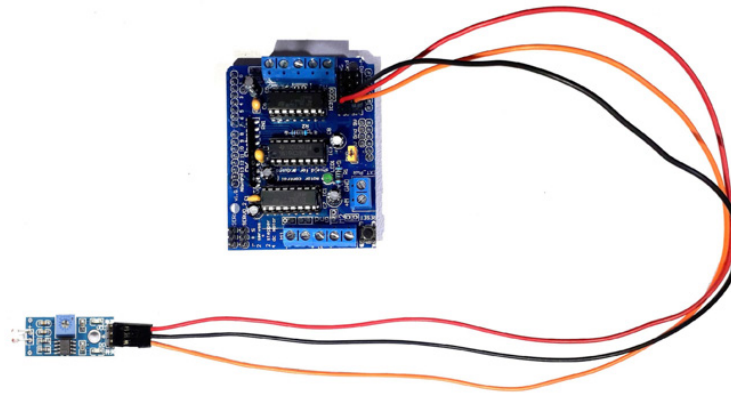
Por ultimo, seguir la siguiente gráfica para colocar el conector en la Shield correctamente.



PASO N° 6

Conexiones del Sensor LDR

(necesario solamente para
la utilización de este sensor)



Conexiones del Sensor LDR.

Este sensor permite medir cantidad de luz, es una resistencia que varía con la llegada de fotones. A mayor luz será mayor la medida, que esta en un rango de 0 a 1023. Cuidado: la conexión puede variar dependiendo el sensor, ya que existen distintos modelos, pero la lógica de conexión es la misma.

PASO N° 6.1.1 Armado de cables.

Crimpeado de cables

En este paso, usaremos unas pinzas de punta plana para crimpear los cables al **Conector Dupont Terminal Hembra 2.54**. Debemos colocar el terminal el 4 cables de los dos lados.

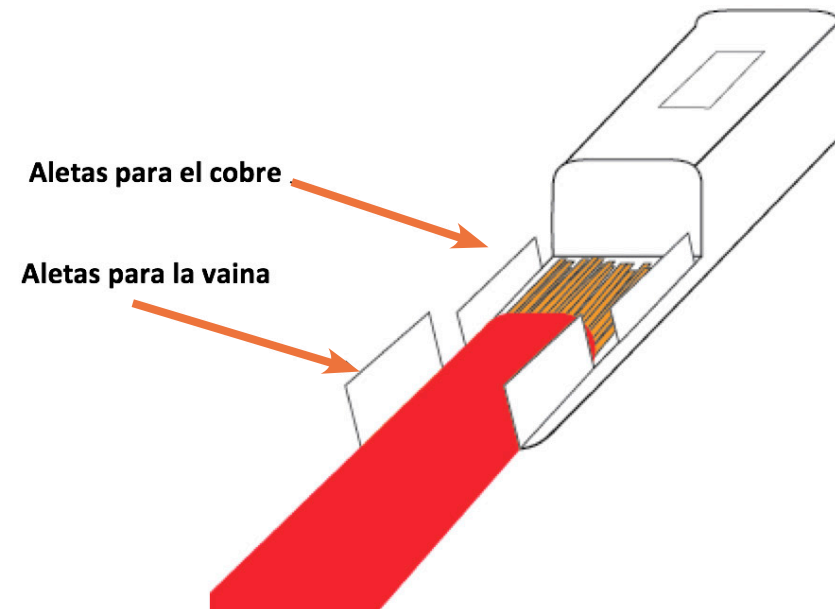


La medida de los cables debe ser de unos 25 cm aproximadamente y en preferencia de distintos colores para identificar facilmente las conexiones.

Crimpeado de cables

La vaina del cable de estas sujeta por unas de las aletas. Y el cobre del cable debe estar apretado por las aletas internas haciendo contacto con el terminal.

PASO N° 6.1.2 Armado de cables.



PASO N° 6.1.3 Armado de cables.

Crimpeado de cables

El cable debe quedar apretado en el terminal firmemente. Y el cobre del metal en contacto con el terminal. Intentar que el cobre del cable no sobresalga por la parte trasera del terminal y que los hilos de cobre entren en el tubo del terminal.

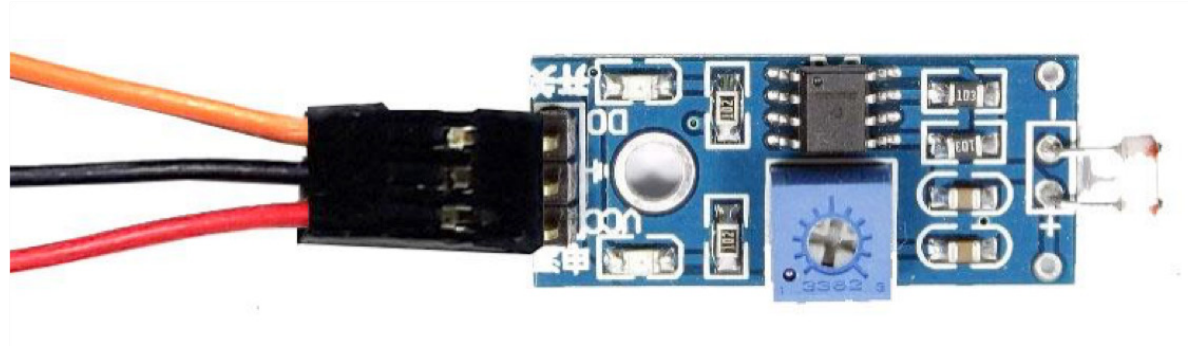
Aletas para el cobre

Aletas para la vaina



PASO N° 6.2

Armado de conectores



Conexión de Cables

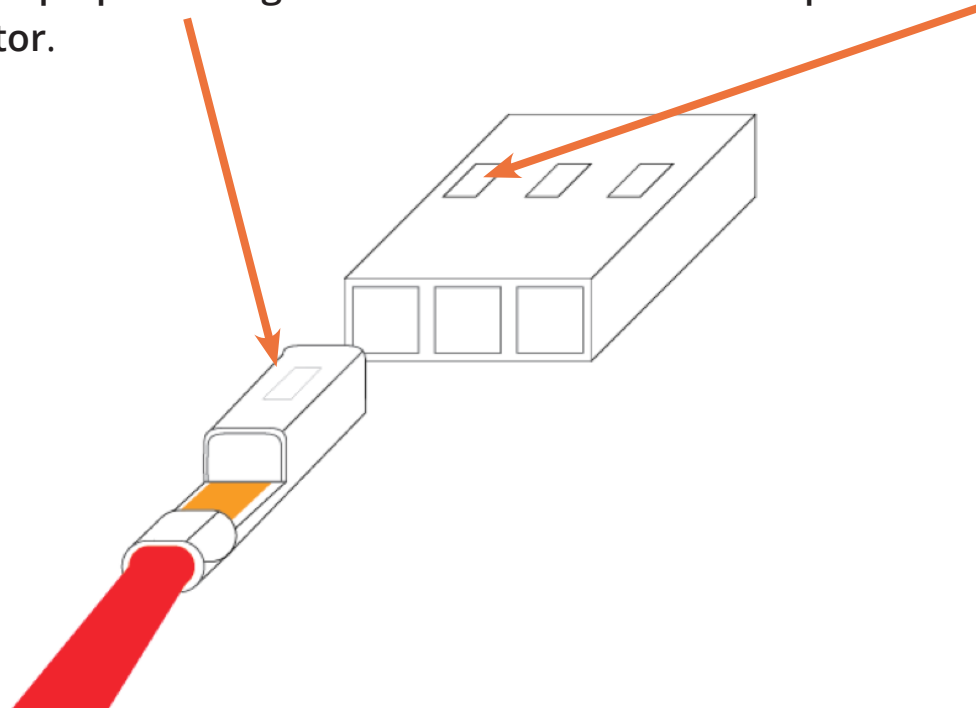
El próximo paso es colocar los cables con los terminales en el conector. Cada cable debe ir en una posición determinada.

ATENCIÓN: Cuidado!!!, si no se conecta adecuadamente es probable que se rompa el sensor o el Arduino.

PASO N° 6.2.1 Armado de conectores

Tira de pines simples.

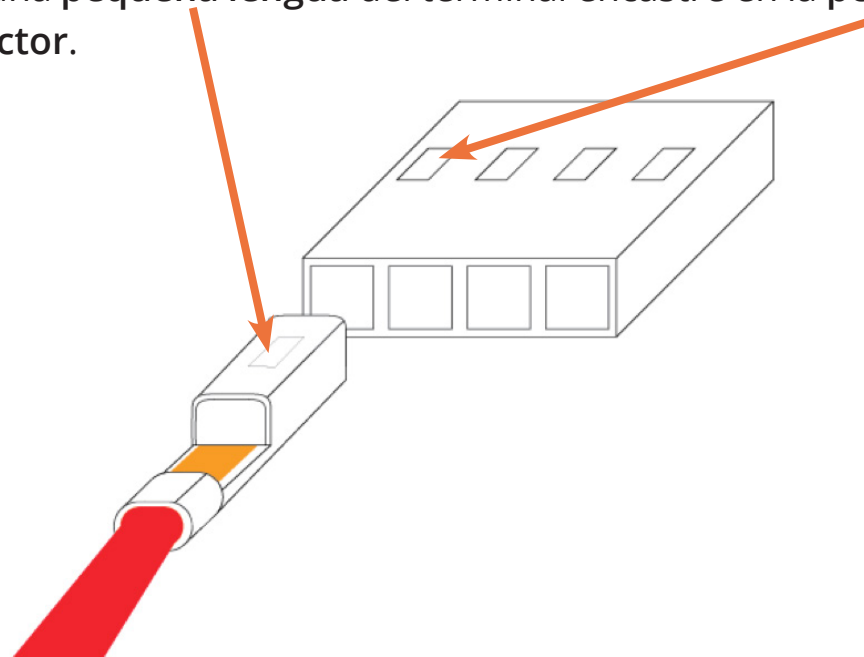
Cortar de la tira de pines una pequeña ficha de solo **tres vías**. Y luego colocar un cable en cada vía del conector. La parte plana del terminal va hacia abajo. No se encuentra la parte plana con la perforación del conector. El terminal entra bien profundamente hasta que una **pequeña lengua** del terminal encastre en la **perforación del conector**.



PASO N° 6.2.1 Armado de conectores

Tira de pines simples.

Hay algunos sensores LDR que poseen **cuatro vías**. En ese caso cortar de la tira de pines una pequeña ficha de solo cuatro vías. Y luego colocar un cable en cada vía del conector. La parte plana del terminal va hacia abajo. No se encuentra la parte plana con la perforación del conector. El terminal entra bien profundamente hasta que una **pequeña lengua** del terminal encastre en la **perforación del conector**.

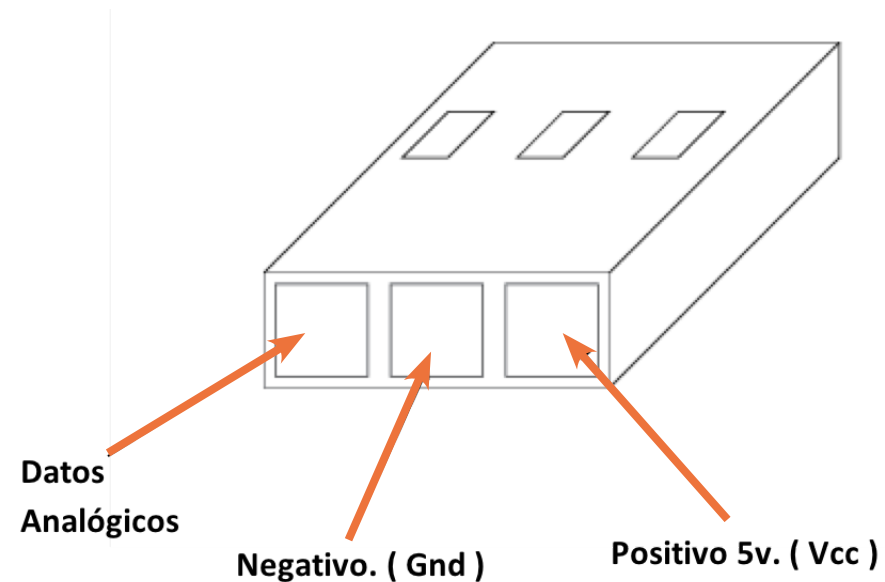


PASO N° 6.2.2

Armado de conectores

Tira de pines simples.

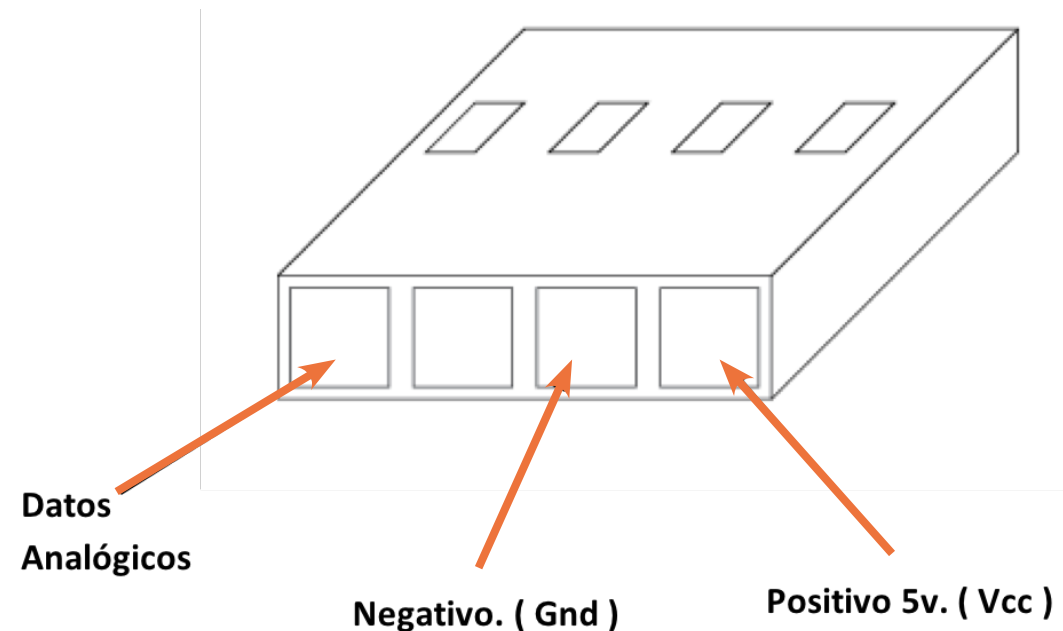
El orden de los cables para el sensor LDR de tres vías es el que se muestra en la imagen. Lo que importa es recordar el color que se utilizó para luego hacer la conexión correcta del otro extremo del cable.



PASO N° 6.2.2 Armado de conectores

Tira de pines simples.

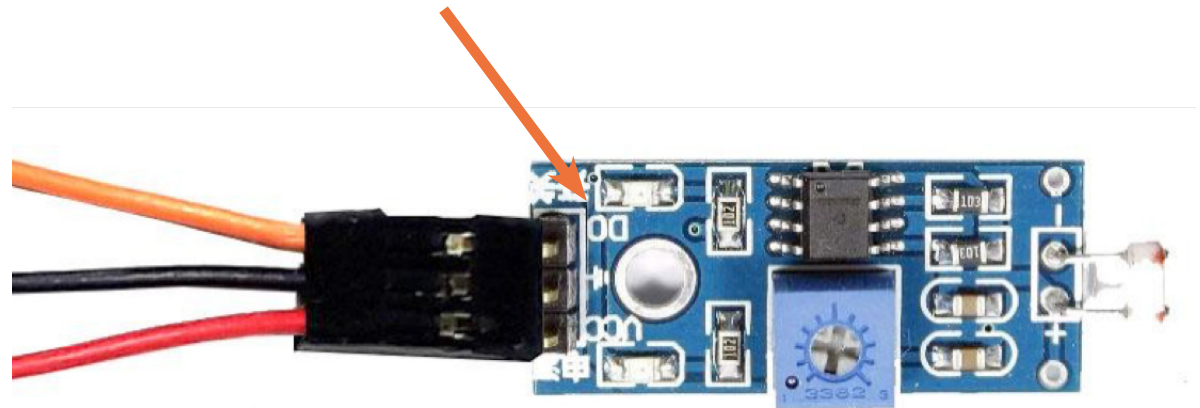
El orden de los cables para el sensor LDR de cuatro vías es el que se muestra en la imagen. El pin D0 no lleva conexión. Lo que importa es recordar el color que se utilizó para luego hacer la conexión correcta del otro extremo del cable.



PASO N° 6.2.3 Armado de conectores

Conectar el cable al sensor ultrasónico.

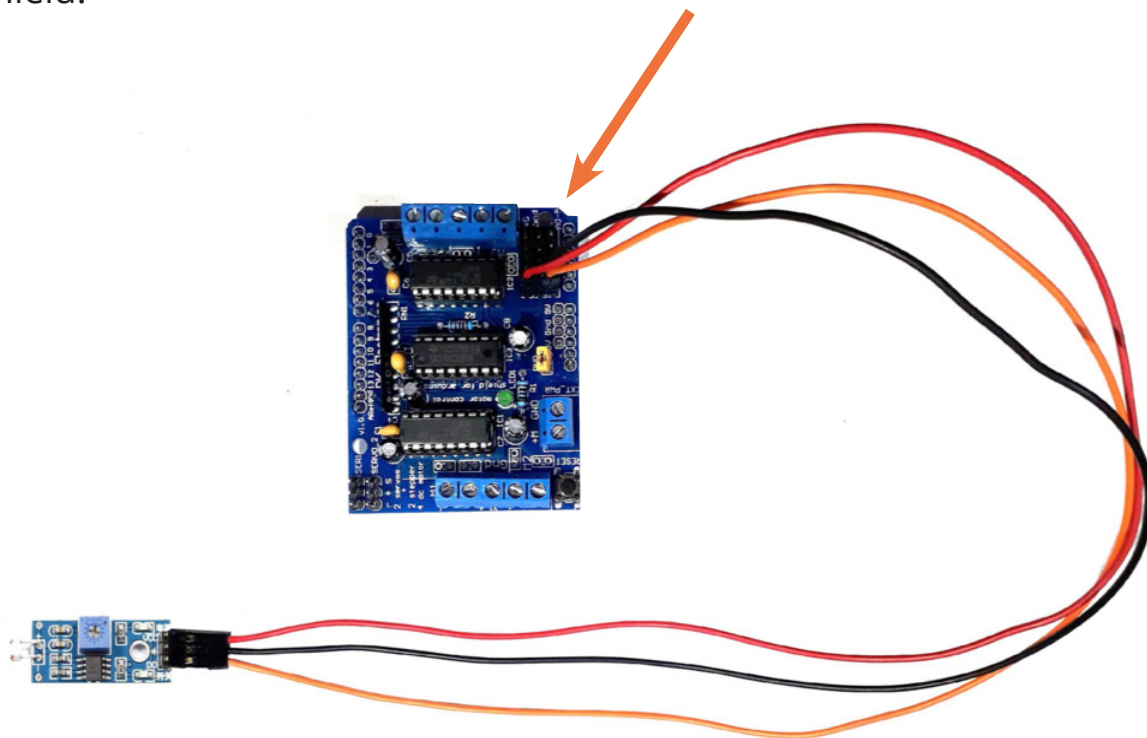
Cuidado!!. Poner el conector de forma que concuerden los cables designados con la inscripción que posee el sensor en cada pin.



PASO N° 6.2.4 Armado de conectores

Conectar los pines hembras en el shield.

Ahora armaremos el conector del cable del sensor LDR del lado del Shield.

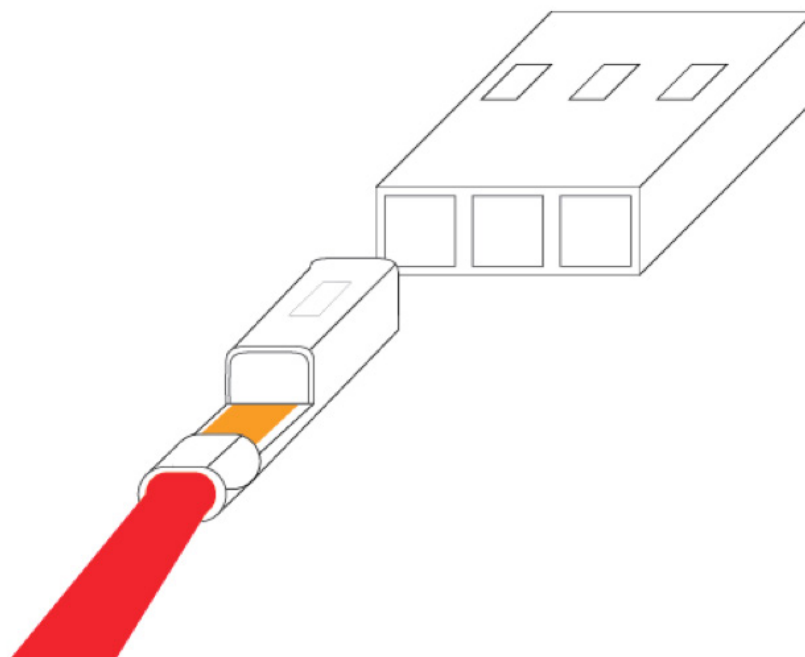


PASO N° 6.2.5

Armado de conectores

Conectar los pines hembras en el shield.

Cortar de la tira de pines en una pequeña ficha de solo **tres vías**. Y luego colocar los tres cables en el conector.

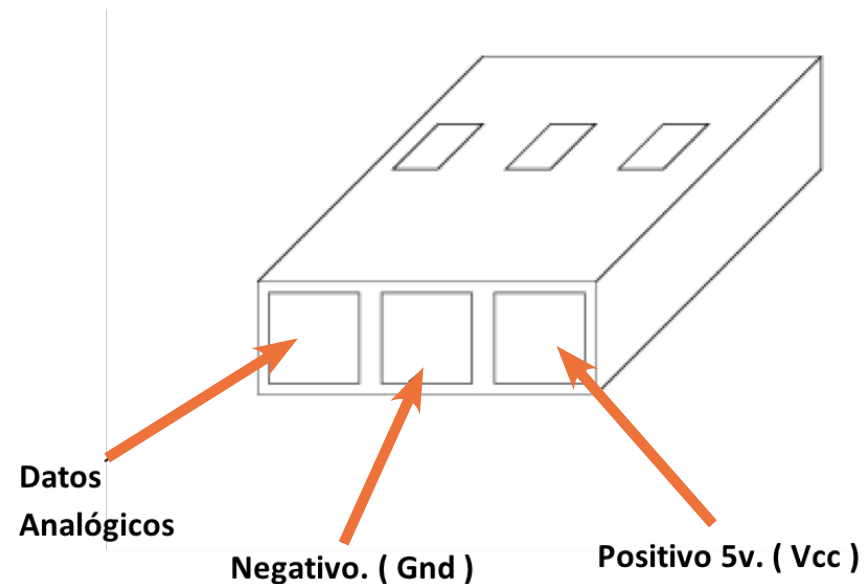


PASO N° 6.2.6 Armado de conectores

Conectar los pines hembras en el shield.

El orden de los cables del lado del Shield esta en el grafico siguiente. Cada cable tiene que respetar la conexión realizada el otro lado.

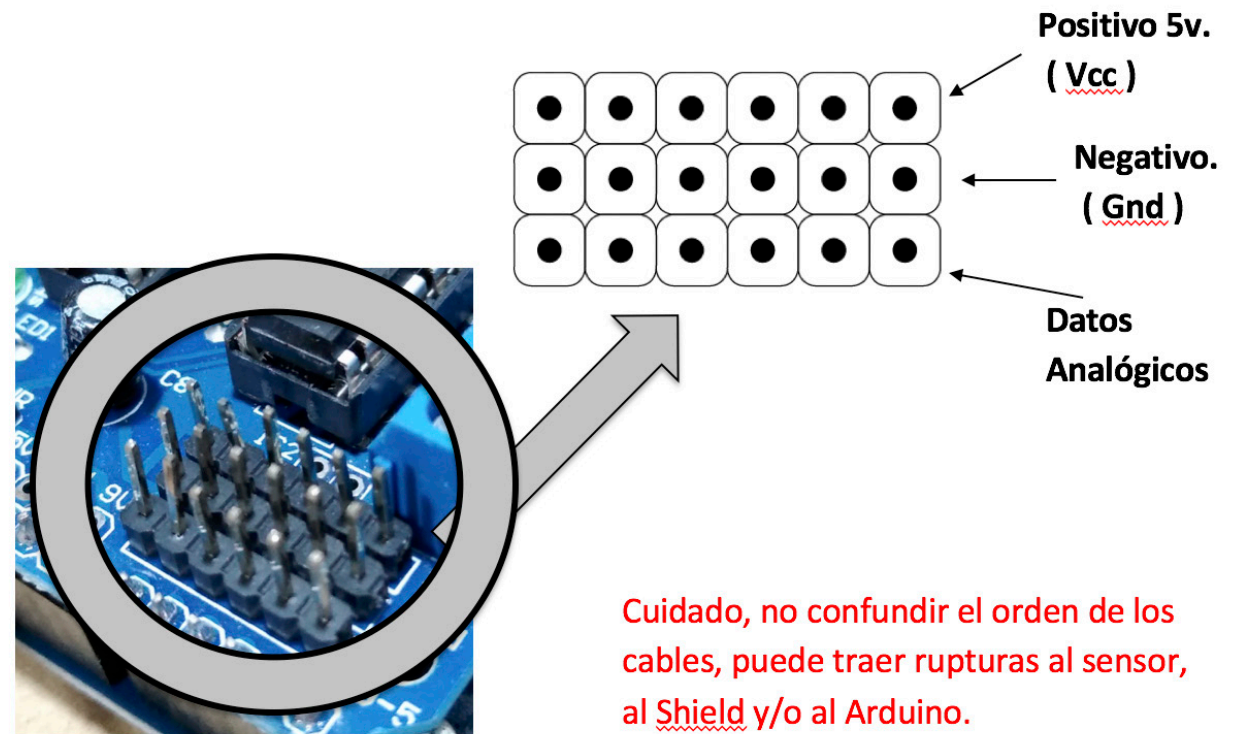
ATENCIÓN: Cuidado, no confundir el orden de los cables, puede traer rupturas al sensor, al Shield y/o al Arduino.



PASO N° 6.2.7 Armado de conectores

Conectar los pines hembras en el shield.

Por último, seguir la siguiente grafica para colocar el conector en la Shield correctamente. En cualquiera de las 6 líneas da igual, solo hay que tener en cuenta su ubicación luego en la programación.



PASO N° 7

Conexiones del Sensor Infrarrojo

(necesario solamente para
la utilización de este sensor)



Conexiones del Sensor Infrarrojo.

Este sensor posee un emisor de luz infrarroja y un receptor de luz infrarroja.

Es módulo que permite medir cuanta luz infrarroja rebota, de su propio emisor, sobre una superficie. A mayor luz de rebote, mayor será la medición, que esta en un rango de 0 a 1023. Por lo general, si la superficie es clara o brillante, rebota mayor cantidad de luz que en una superficie oscura o rugosa.

ATENCIÓN: Cuidado! la conexión puede variar dependiendo el modelo de sensor, ya que existen varios, pero la lógica de conexión es la misma.

PASO N° 7.1.1 Armado de cables.

Crimpeado de cables

En este paso, usaremos unas pinzas de punta plana para crimpear los cables al **Conector Dupont Terminal Hembra 2.54**. Debemos colocar los terminales a 3 cables de ambos lados.

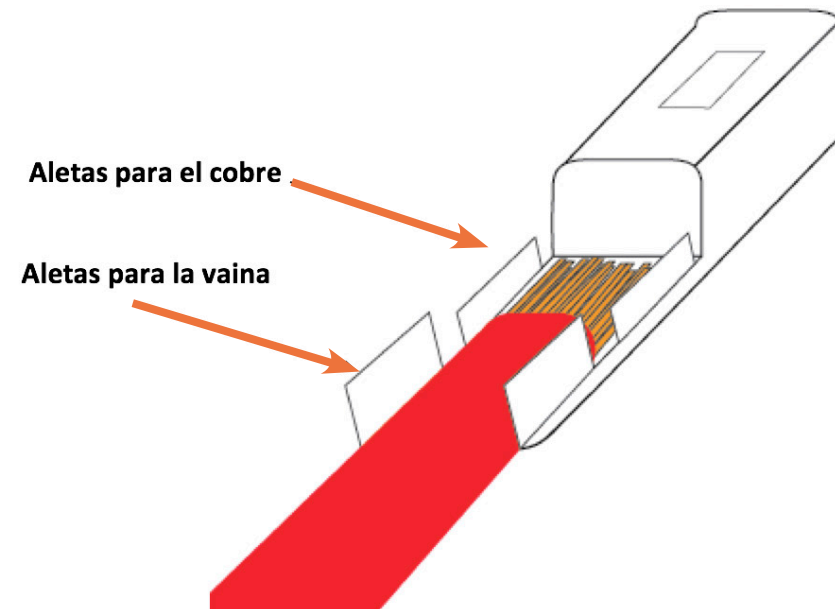


La medida de los cables debe ser de unos 25 cm aproximadamente y en preferencia de distintos colores para identificar fácilmente las conexiones.

Crimpeado de cables

La vaina del cable de estas sujeta por unas de las aletas. Y el cobre del cable debe estar apretado por las aletas internas haciendo contacto con el terminal.

PASO N° 7.1.2 Armado de cables.



PASO N° 7.1.3 Armado de cables.

Crimpeado de cables

El cable debe quedar apretado en el terminal firmemente. Y el cobre del metal en contacto con el terminal. Intentar que el cobre del cable no sobresalga por la parte trasera del terminal y que los hilos de cobre entren en el tubo del terminal.

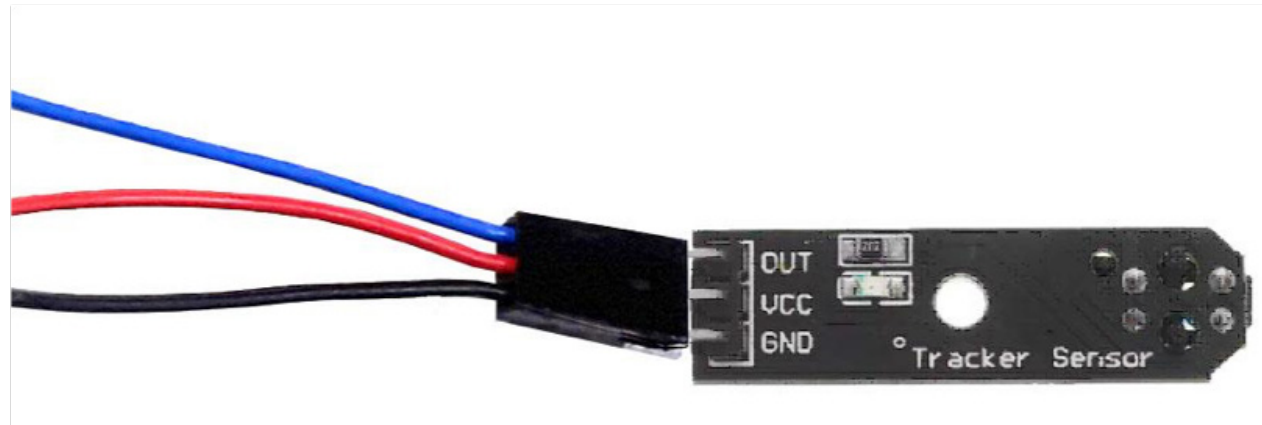
Aletas para el cobre

Aletas para la vaina



PASO N° 7.2

Armado de conectores



Conexión de Cables

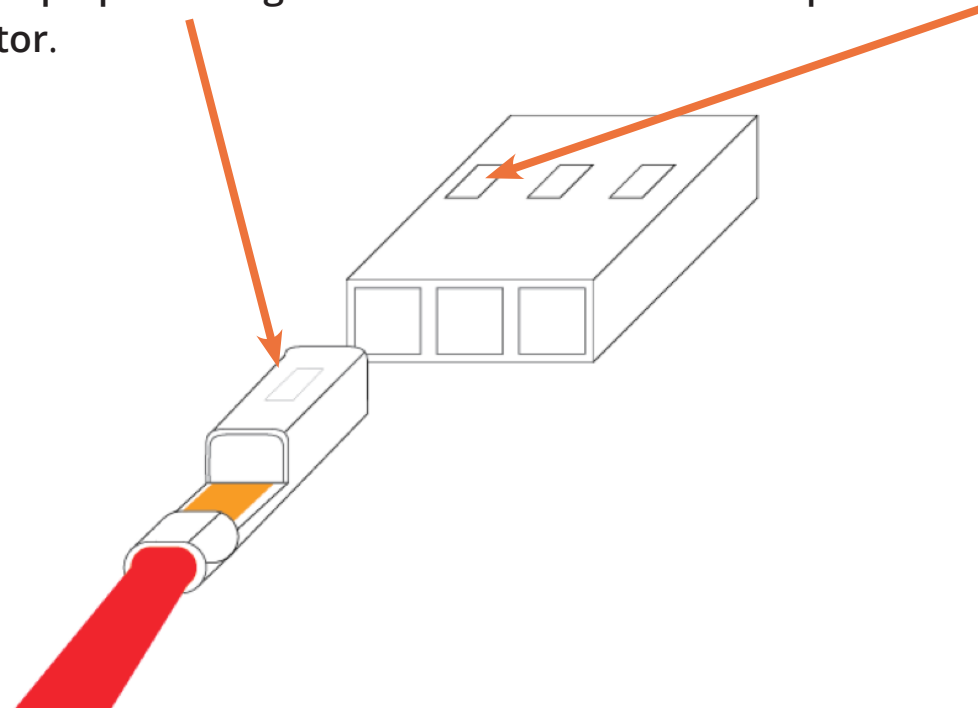
El próximo paso es colocar los cables con los terminales en el conector. Cada cable debe ir en una posición determinada.

ATENCIÓN: Cuidado!!!, si no se conecta adecuadamente es probable que se rompa el sensor o el Arduino.

PASO N° 7.2.1 Armado de conectores

Tira de pines simples.

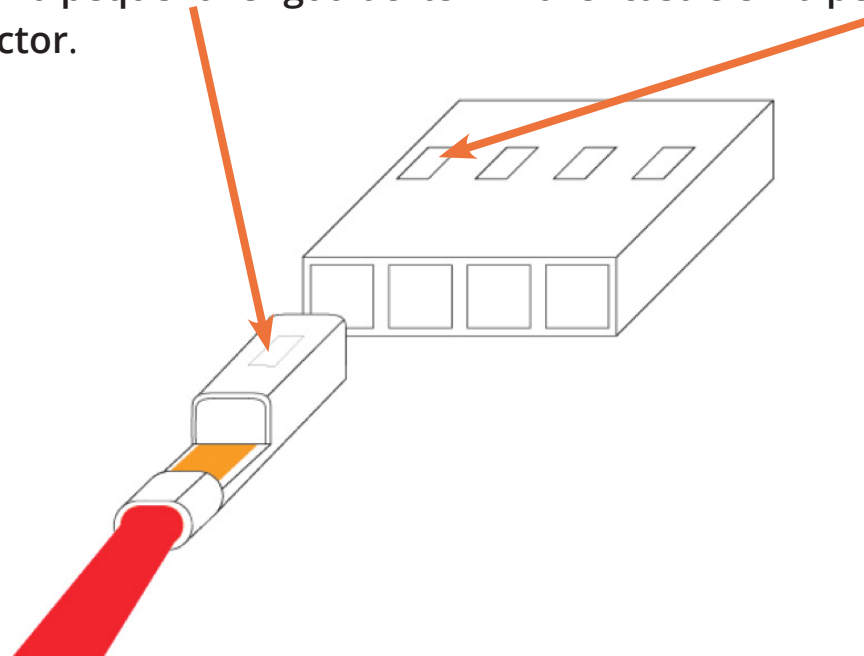
Cortar de la tira de pines una pequeña ficha de solo **tres vías**. Y luego colocar un cable en cada vía del conector. La parte plana del terminal va hacia abajo. No se encuentra la parte plana con la perforación del conector. El terminal entra bien profundamente hasta que una **pequeña lengua** del terminal encastre en la **perforación del conector**.



PASO N° 7.2.1 Armado de conectores

Tira de pines simples.

Hay algunos sensores infrarrojos que poseen **cuatro vías**. En ese caso cortar de la tira de pines una pequeña ficha de solo cuatro vías. Y luego colocar un cable en cada vía del conector. La parte plana del terminal va hacia abajo. No se encuentra la parte plana con la perforación del conector. El terminal entra bien profundamente hasta que una **pequeña lengua** del terminal encastre en la **perforación del conector**.

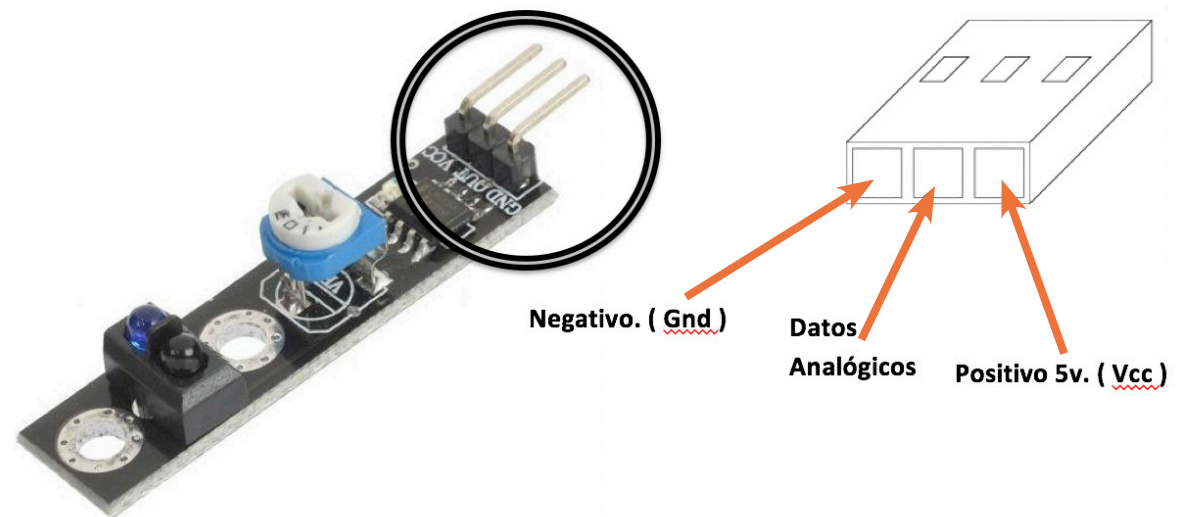


PASO N° 7.2.2

Armado de conectores

Tira de pines simples.

El orden de los cables para el sensor Infrarrojo de **tres vías** varía según el modelo. Puede ser el siguiente. Es importante mirar las inscripciones que posee el sensor para armar el conector. También es importante recordar el color que se utilizó en cada cable para luego hacer la conexión correcta del otro extremo del cable.

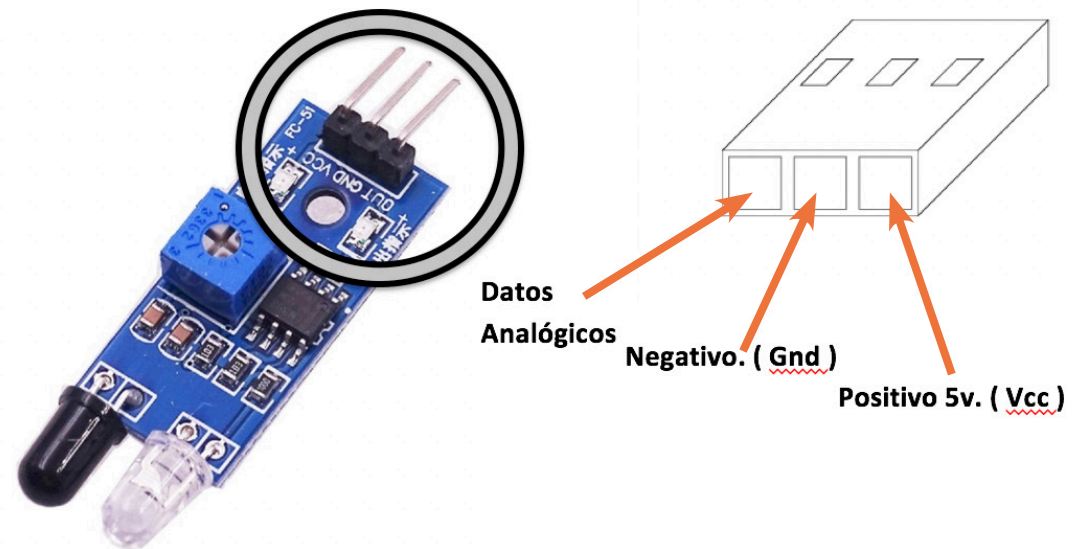


PASO N° 7.2.2

Armado de conectores

Tira de pines simples.

El orden de los cables para el sensor Infrarrojo de **tres vías** varía según el modelo. Puede ser el siguiente. Es importante mirar las inscripciones que posee el sensor para armar el conector. También es importante recordar el color que se utilizó en cada cable para luego hacer la conexión correcta del otro extremo del cable.

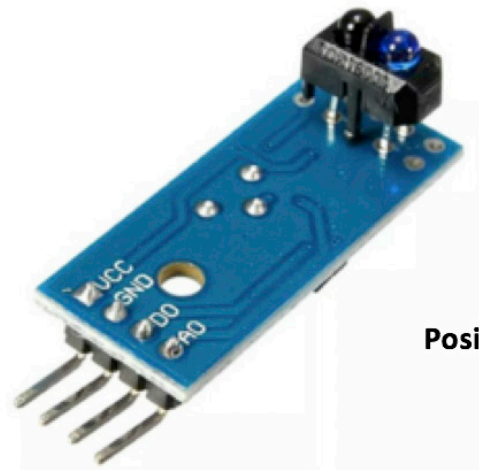


PASO N° 7.2.2

Armado de conectores

Tira de pines simples.

El orden de los cables para el sensor Infrarrojo de **cuatro vías** es el siguiente. El pin D0 no lleva conexión. Lo que importa es recordar el color que se utilizó para luego hacer la conexión correcta del otro extremo del cable.



Positivo 5v. (Vcc)

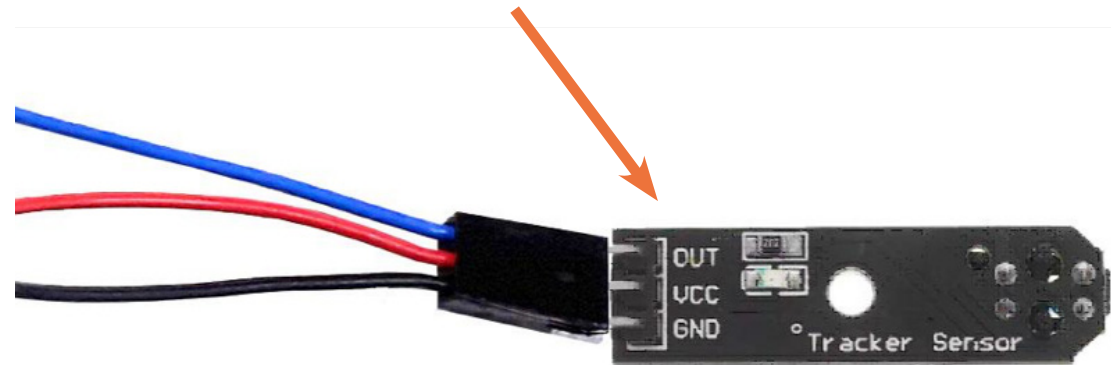
Negativo. (Gnd)

Datos
Analógicos

PASO N° 6.2.3 Armado de conectores

Conectar el cable al sensor infrarrojo.

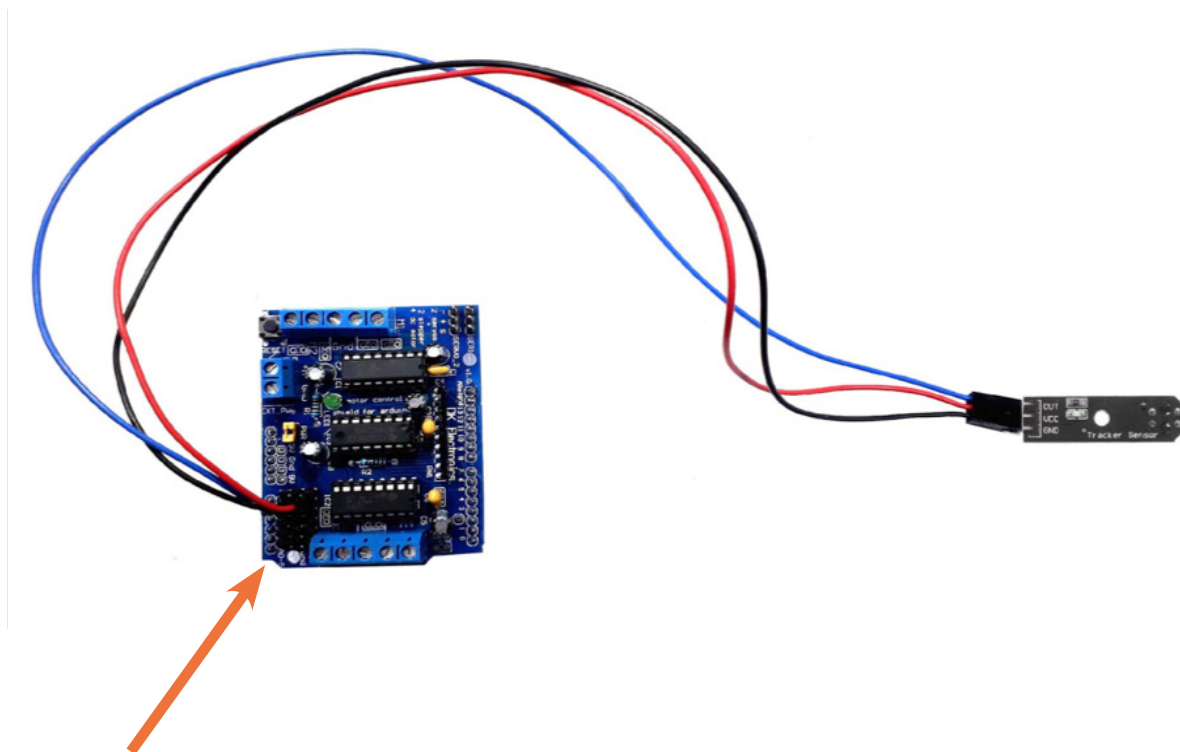
Cuidado!!. Poner el conector de forma que concuerden los cables designados con la inscripción que posee el sensor en cada pin.



PASO N° 7.2.4 Armado de conectores

Conectar los pines hembras en el shield.

Ahora armaremos el conector del cable del sensor Infrarrojo del lado del Shield.

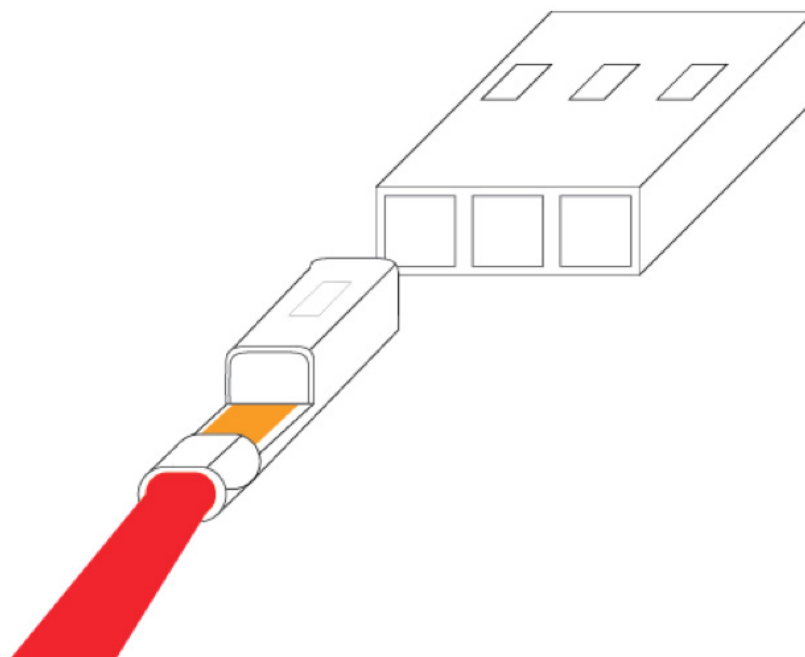


PASO N° 7.2.5

Armado de conectores

Conectar los pines hembras en el shield.

Cortar de la tira de pines en una pequeña ficha de solo **tres vías**. Y luego colocar los tres cables en el conector.

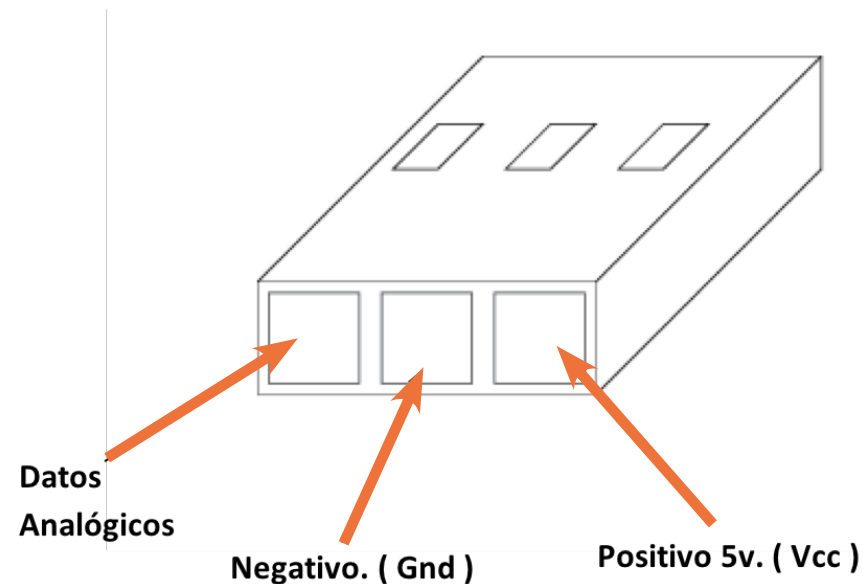


PASO N° 7.2.6 Armado de conectores

Conectar los pines hembras en el shield.

El orden de los cables del lado del Shield esta en el grafico siguiente. Cada cable tiene que respetar la conexión realizada el otro lado.

ATENCIÓN: Cuidado, no confundir el orden de los cables, puede traer rupturas al sensor, al Shield y/o al Arduino.

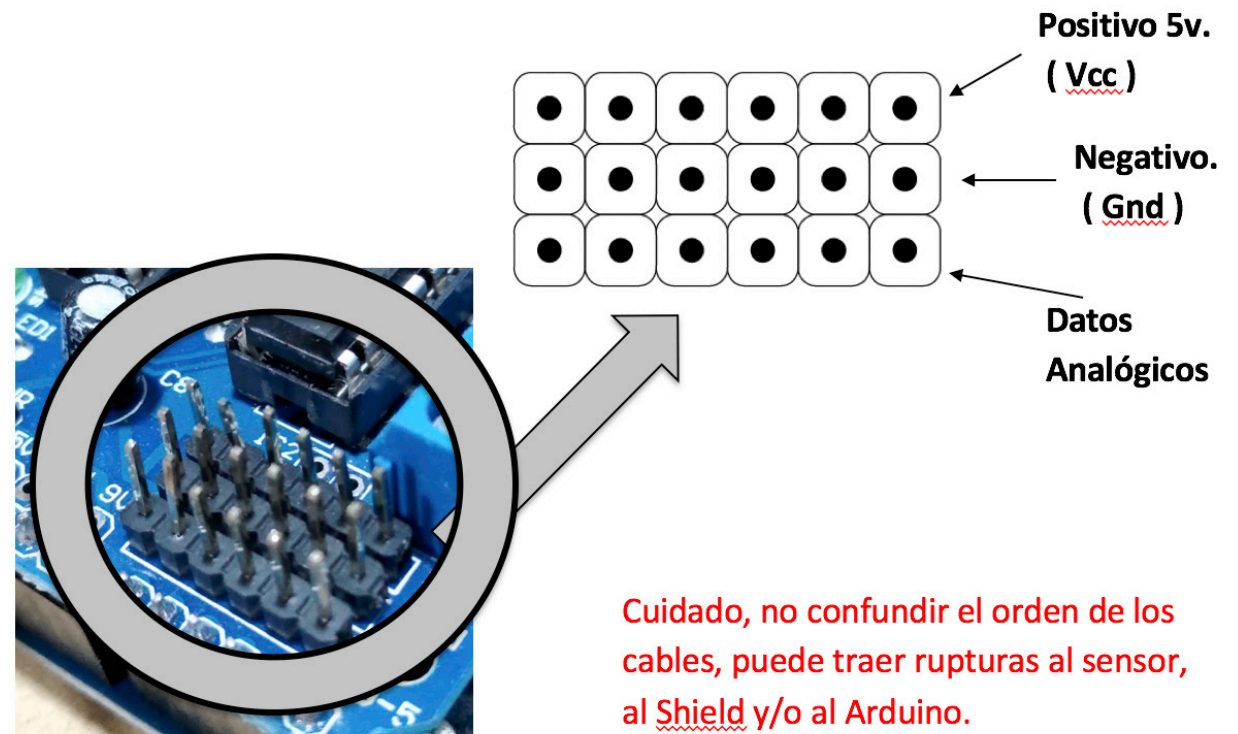


PASO N° 7.2.7

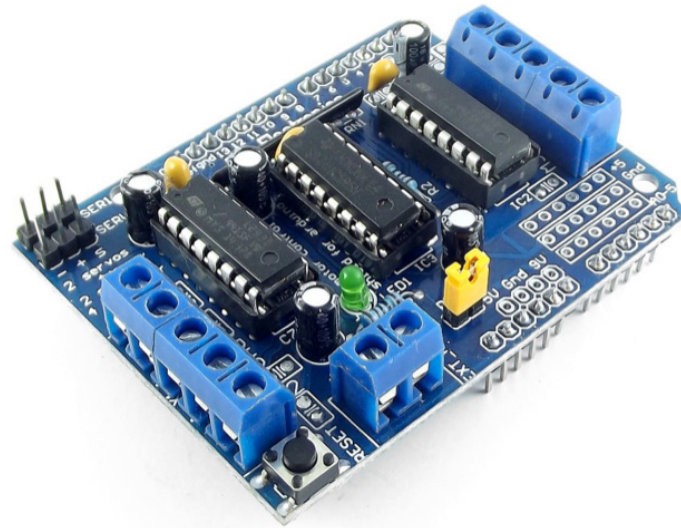
Armado de conectores

Conectar los pines hembras en el shield.

Por último, seguir la siguiente grafica para colocar el conector en la Shield correctamente. En cualquiera de las 6 líneas da igual, solo hay que tener en cuenta su ubicación luego en la programación.



PASO N° 8 CONEXIÓN DE BATERÍAS



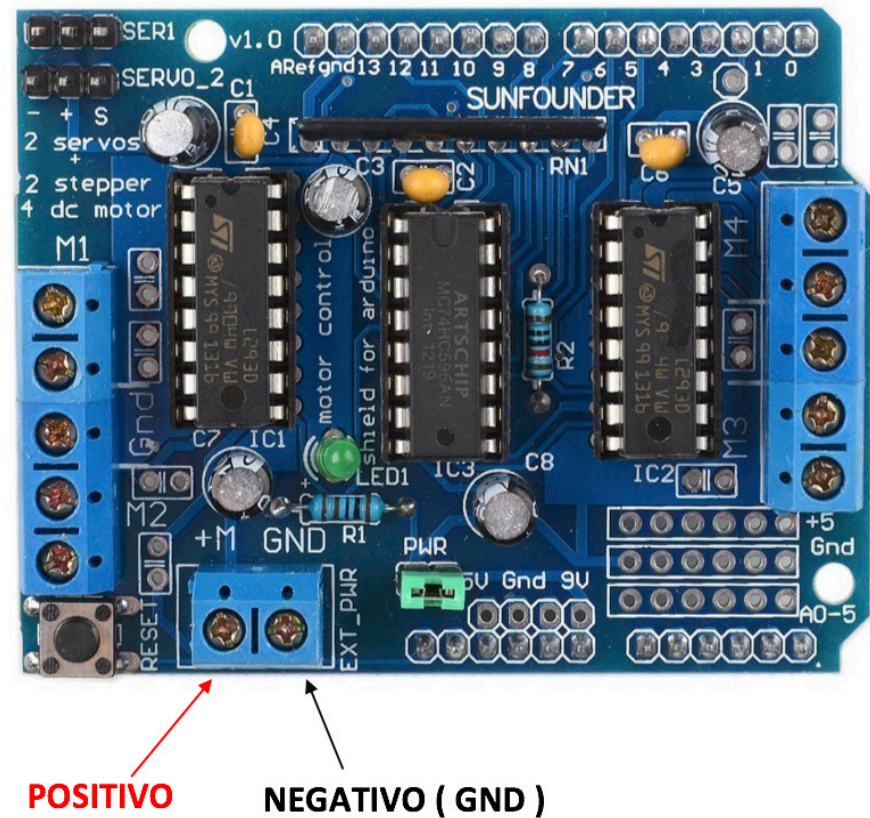
Conexiones de las Bateria.

Este paso es muy importante, **no hay que confundir la conexión porque hay peligro de ruptura**. La conexión se realiza en las bornes correctas sin confundir el **positivo** y el **negativo**. El negativo está marcado con la sigla GND: Los voltajes que soporta es de 5 a 19 volt. Nuestra propuesta es trabajar con dos baterías de 3.7. **Un total de 7.4 volt máximo.**

Conexiones de las Baterías.

PASO N° 8.1

Conexiones de la batería

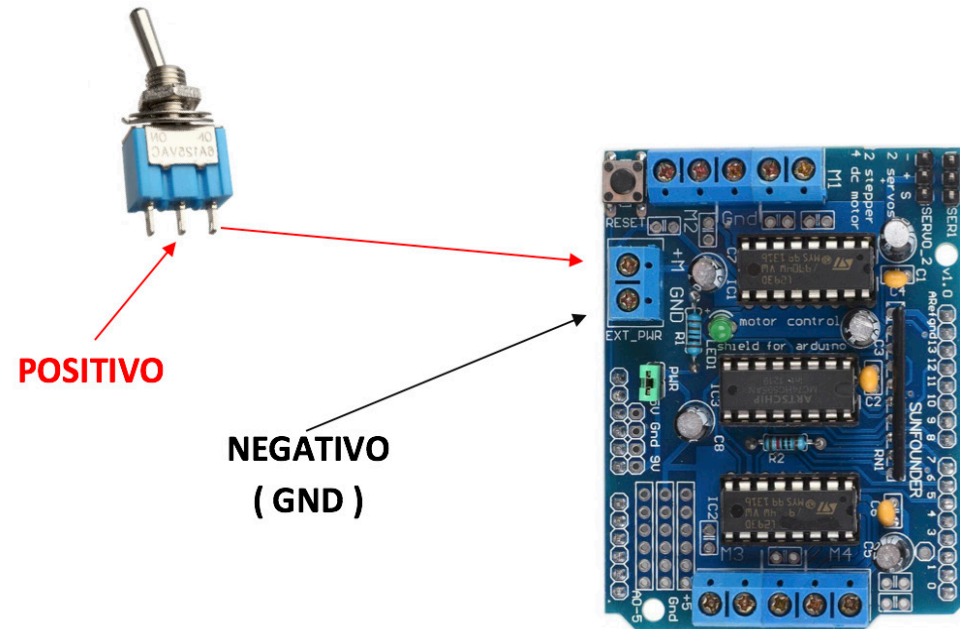


PASO N° 8.2

Conexiones de la batería

Conexiones de las Baterías.

Tener en cuenta que al realizar la conexión el robot arrancará. Si posee un programa anterior cargado se ejecutará y empezará a moverse. Para evitar esto es conveniente colocar un interruptor en el conector. Así se puede prender y apagar el robot desde allí.



¿Cómo programar ComPacto?

El robot **ComPacto** posee una placa **Arduino Uno** como controlador de sus sistemas. El programa que realizaremos para controlar el robot debe alojarse en la placa, a esa acción la llamaremos **cargar el programa**.

El programa debe escribirse en lenguaje **Arduino** desde su propio **Ide** (entorno de desarrollo integrado). El **Ide** es un procesador de texto dedicado al lenguaje, que con algunas herramientas nos ayuda a programar y cargar el programa al robot.

El lenguaje **Arduino** puede ser complejo para los recién iniciados en la programación, por ello **ComPacto** posee su propio **Ide** que permite acercarse a la programación de forma visual y más intuitiva. El **Ide** de **ComPacto** generará un código **Arduino** que debe abrirse con el **Ide** de **Arduino** para ser cargado en el robot. Este paso es importante, el objetivo es que el programador de a poco se vaya familiarizando con el lenguaje **Arduino** , que en el fondo es realmente la forma de programar eficientemente.

Resumiendo, para poder programar **ComPacto** hay que descargar de internet el Ide de **Arduino** y el **Ide** de **ComPacto**.

El programa lo realizaremos en el **Ide** de **Compacto**, lo guardaremos en el disco rígido, luego lo abriremos con el **Ide** de **Arduino** para poder cargárselo al robot. Este procedimiento parece a primera vista engorroso pero al utilizarlo no es nada incómodo.

La guía de instalación del **Ide** de **Arduino** se encuentra en un anexo al final de este tutorial.

Ide de ComPacto

Descarga e instalación

El **Ide** (entorno de desarrollo integrado) de **ComPacto** se encuentre las descargas de:

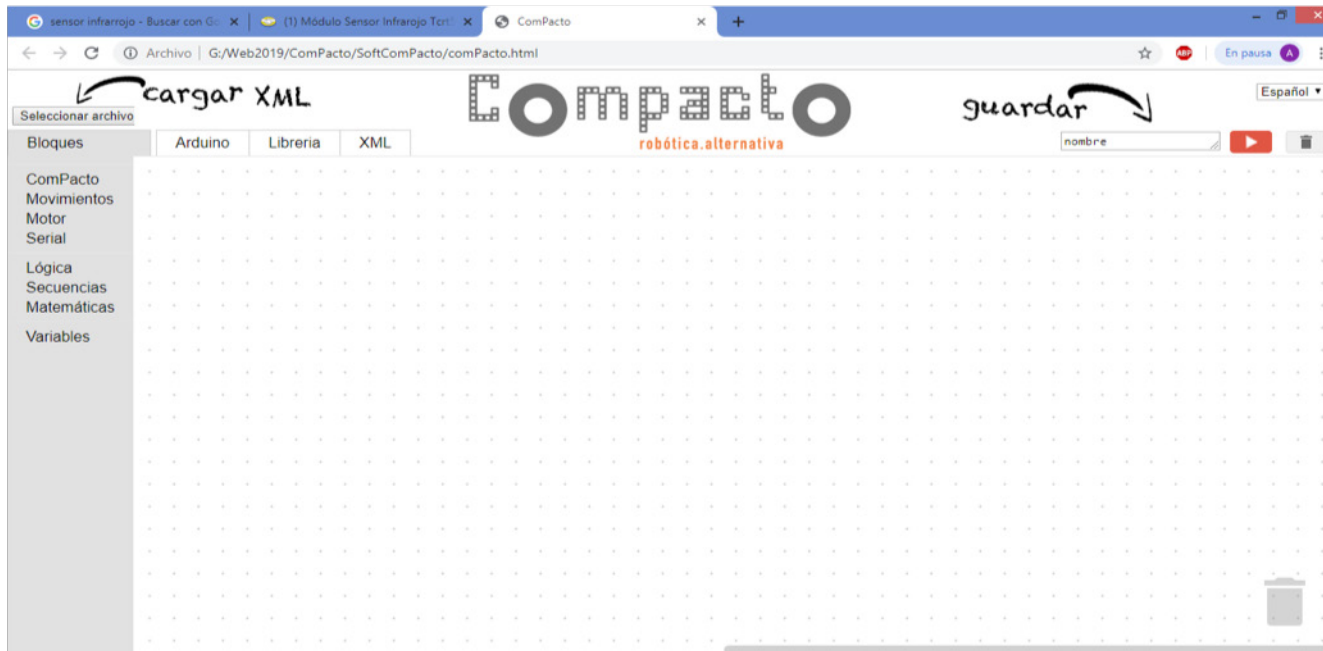
<http://www.roboticaalternativa.com/comPacto.html>

Se descargara el archivo **compacto.zip** que hay que descomprimir. Dentro de los archivos descomprimidos, se encontrara un carpeta con el nombre **“softComPacto”**.

Dentro de esta se encontrara el archivo **“comPacto.html”**. Este archivo es el **Ide** de **“Compacto”**.

Ide de ComPacto

El **Ide** (entorno de desarrollo integrado) de “**Compacto**” es un archivo HTML, en otras palabras, es una pagina web. Puede utilizarse con cualquier navegador web, pero se recomienda utilizar [Google Chrome](#).

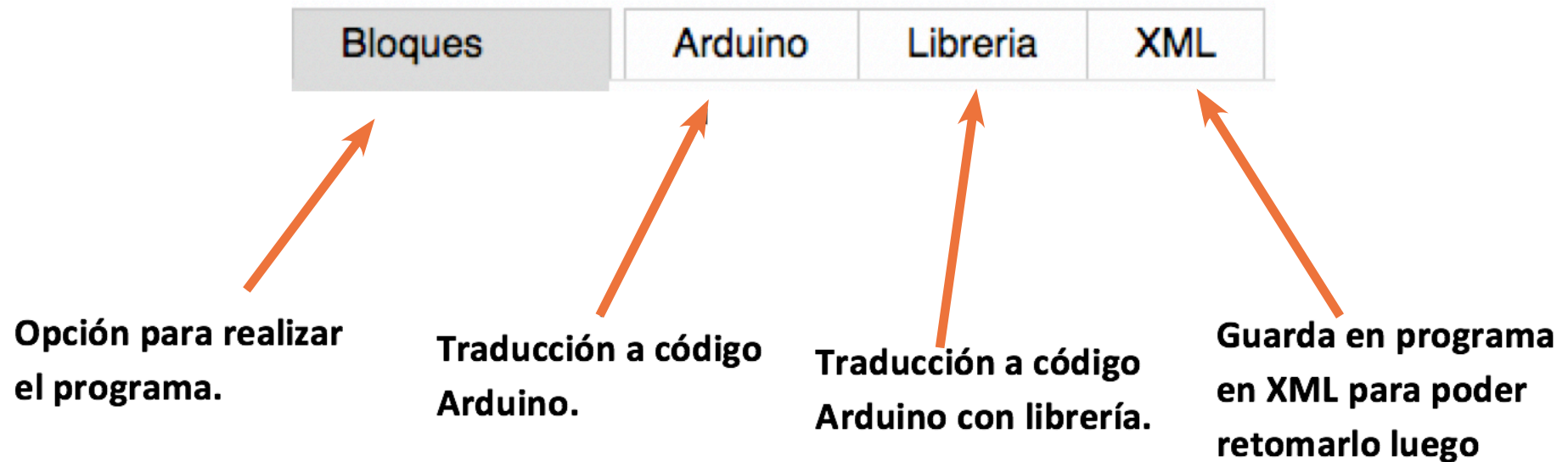


El **Ide** consta de un área de trabajo blanca punteada, un menú horizontal superior y un menú vertical a la izquierda.

En este menú se encuentran todos los bloques que podemos utilizar agrupados por su función. El menú horizontal superior contiene las opciones de traducción del programa y guardado

Menú horizontal

El **menú horizontal** contiene la opciones de traducción de lenguajes. El programa se realizara en la opción “Bloques” y al tocar cualquiera de las otras pestaña, este programa de bloques se traducirá a la opción seleccionada.





Menú horizontal Arduino

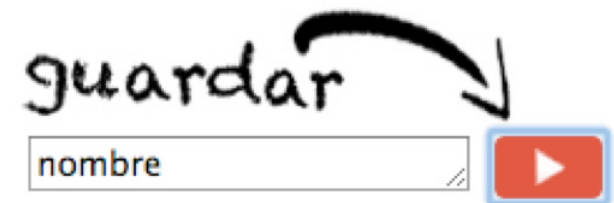
Al seleccionar esta pestaña, el programa realizado con bloques pre-diseñados, se traducirá instantáneamente a **código Arduino**. Si se modifica un bloque al volver a esta pestaña se cambiara el código automáticamente.

Para trasladar este código al **Ide de Arduino** y finalmente cargárselo al robot, se puede tomar dos caminos.

#1 El primero es seleccionar el texto de código con el Mouse y copiarlo con las teclas "**control+c**" y pegarlo en el **Ide de Arduino** con las teclas "**control+v**".

#2 El otro camino es descargar el código presionando el botón "**guardar**" luego de colocarle el nombre.

Se guardara un archivo .ino (.arduino) con el nombre seleccionado más los datos del día y horario actual. Esto es para que no se superpongan versiones del mismo proyecto. Luego se podrá abrir con el Ide Arduino.





Menú horizontal Arduino con Librería

Al seleccionar esta pestaña, el programa realizado con bloques ya pre-diseñados, se traducirá instantaneamente a **código Arduino con librería**. Es decir que los comandos serán funciones que ocultan parte del lenguaje **Arduino** pero es más sencillo de leer. Si se modifica un bloque al volver a esta pestaña se cambiará el código automáticamente.

Para la utilización del código con librería, previamente hay que instalar la librería.

La librería se encuentra entre los documentos descargados, se trata de la carpeta que lleva el nombre de **LibreriaComPacto**. Para instalar la librería solamente hay que copiar esta carpeta a **mis-Documentos/Arduino/libraries** de tu computadora (previamente hay que instalar el **Ide de Arduino** y ejecutarlo al menos una vez).

Una vez copiado, hay que cerrar el **Ide de Arduino** y volver a abrirlo (al iniciar Arduino nuevamente se reconocerá sus librerías nuevas).



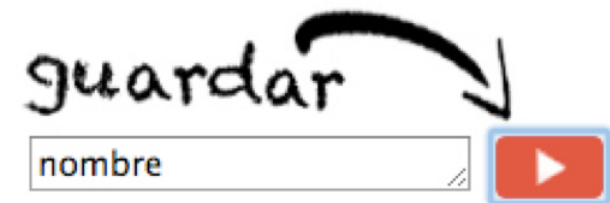
Menú horizontal Arduino con Librería

Para trasladar este código al **Ide de Arduino** y finalmente cargárselo al robot, se puede tomar dos caminos.

#1 El primero es seleccionar el texto de código con el Mouse y copiarlo con las teclas “**control+c**” y pegarlo en el **Ide de Arduino** con las teclas “**control+v**”.

#2 El otro camino es descargar el código presionando el botón “guardar” luego de colocarle el nombre.

Se guardara un archivo .ino (.arduino) con el nombre seleccionado más los datos del día y horario actual. Esto es para que no se superpongan versiones del mismo proyecto. Luego se podrá abrir con el **Ide Arduino**.



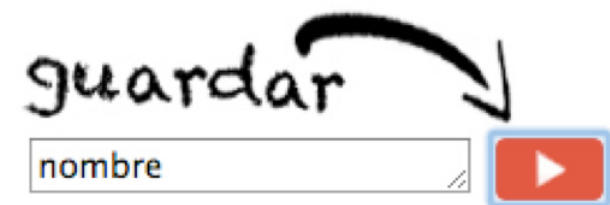


Menú horizontal XML

Al seleccionar esta pestaña, el programa realizado con bloques pre-diseñados, se traducirá instantáneamente a **código XML**. Si se modifica un bloque al volver a esta pestaña se cambiará el código automáticamente.

Esta pestaña permite guardar el proyecto en forma de bloques, para poder recuperar el programa y seguir trabajando sobre él. Para guardar hay que presionar el botón **"guardar"** luego de colocarle el nombre.

Se guardará un archivo .xml con el nombre seleccionado más los datos del día y horario actual. Esto es para que no se superpongan versiones del mismo proyecto. Luego se podrá abrir con el **Ide Arduino**.



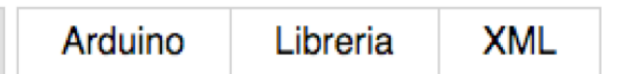
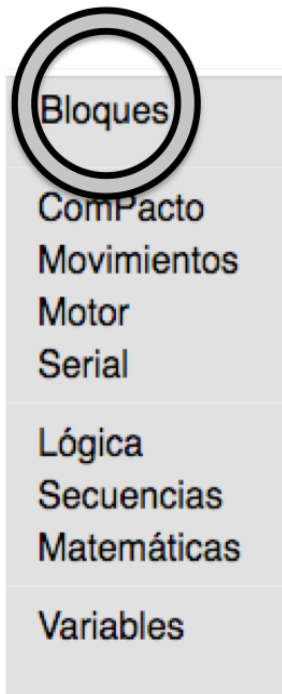


Menú horizontal XML

Para recuperar un programa ya guardado en **xml** hay que presionar el botón “**cargar xml**”.

Luego en la ventana emergente seleccione el programa deseado. Este proceso recuperará el programa de bloques guardado.





Menú horizontal Bloques

En este menú se presentan todos los bloques que posee “ComPacto” para poder programar el robot.

Todos los bloque se encuentran agrupados en sub menús según su función en el menú vertical izquierdo.

Para programar solamente hay que buscar el bloque deseado en los menús.

Una vez encontrado hay que arrastrarlo con el mouse sobre el lienzo blanco punteado de trabajo.

Para borrar un bloque del lienzo de trabajo, hay que seleccionarlo con el mouse y arrastrarlo al tacho de **basura**



Menú horizontal Bloques

Bloques	Arduino	Libreria	XML
ComPacto			
Movimientos			
Motor			
Serial			
Lógica			
Secuencias			
Matemáticas			
Variables			

En menú **"ComPacto"** se encuentran los bloques con funciones del robot que no sean movimientos. Por ejemplo las funciones de sensado, de espera y de no repetición de ciclos.

En el menú **"Movimientos"** están las funciones de movimientos del robot sin diferenciar los motores. Por ejemplo el avanzar o retroceder los dos motores simultáneamente.

El menú **"Motor"** podemos encender y apagar los motores en forma independiente entre ellos.

Menú horizontal Bloques

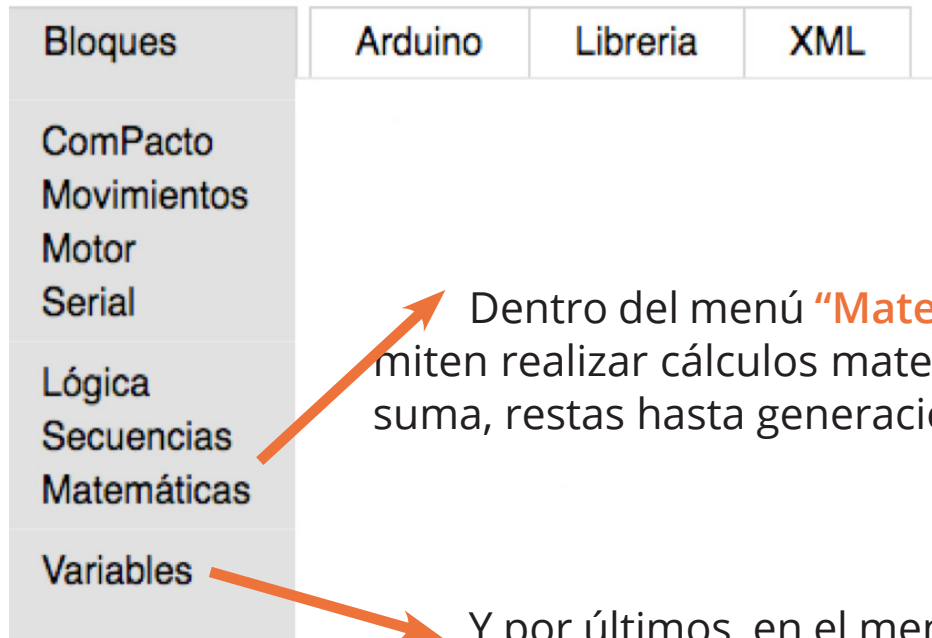
Bloques	Arduino	Libreria	XML
ComPacto			
Movimientos			
Motor			
Serial			
Lógica			
Secuencias			
Matemáticas			
Variables			

El menú **"Serial"** es más avanzado, tiene la función de comunicación entre el robot y la computadora en tiempos de ejecución. Por ejemplo, puedo comunicar al robot con la computadora para ver los valores de los sensores en tiempo real durante la ejecución de un programa.

El menú de **"Lógica"** contiene las herramientas necesarias para la toma de decisiones de un programa. Por ejemplo si un robot descubre a través de sus sensores que se encuentra un obstáculo al frente, este robot puede decidir retroceder y girar para evitarlo.

En **"Secuencias"** están los bloques de loops. Estos sirven para realizar tareas repetitivas. Por ejemplo, si deseo que mi robot repita 10 veces el mismo movimiento, entre estos bloques encontraré las herramientas para esto, sin tener que colocar 10 veces los mismos bloques.

Menú horizontal Bloques



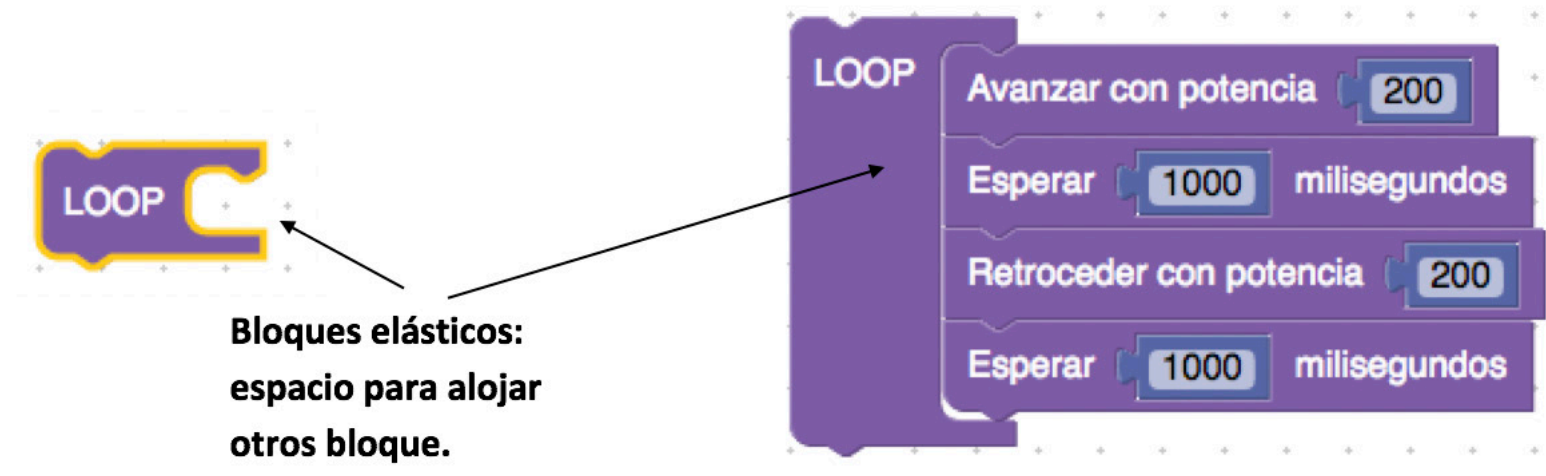
Dentro del menú **“Matemáticas”** encontraré una serie de bloques que me permiten realizar cálculos matemáticos. Algunos simples y otro más complejos. Como suma, restas hasta generación de números aleatorios entre otros.

Y por últimos, en el menú **“Variables”** encontrarás la posibilidad de la creación de variables. Son espacio de memorias que se van a utilizar cuando se quiere que el programa recuerde valores. Por ejemplo, registrar los valores de un sensor para luego poder hacer una análisis del mismo.

Tipos de Bloques

Existen distintos tipos de bloques. Algunos bloques contienen un espacio para alojar otros bloques para su funcionamiento, los que se denominan parámetros.

Y hay otros bloques que los llamaremos **elásticos**, a parte de los parámetros de funcionamiento, pueden estirarse para alojar más bloques.



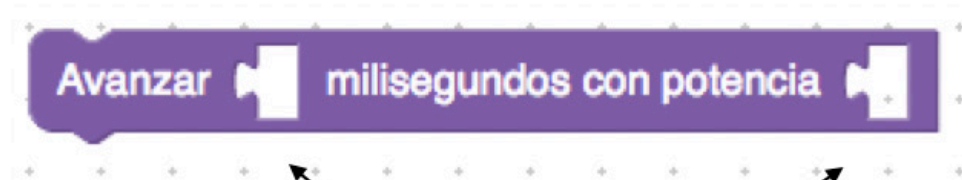
Los bloques son intuitivos y a simple vista puede verse en su morfología estas diferencias.

Entre bloques, se encastran con unas **anclas** imantadas en forma lineal. Uno debajo del otro.

Ejemplo de anclaje entre bloques.



Ejemplo de Parámetros de los bloques.

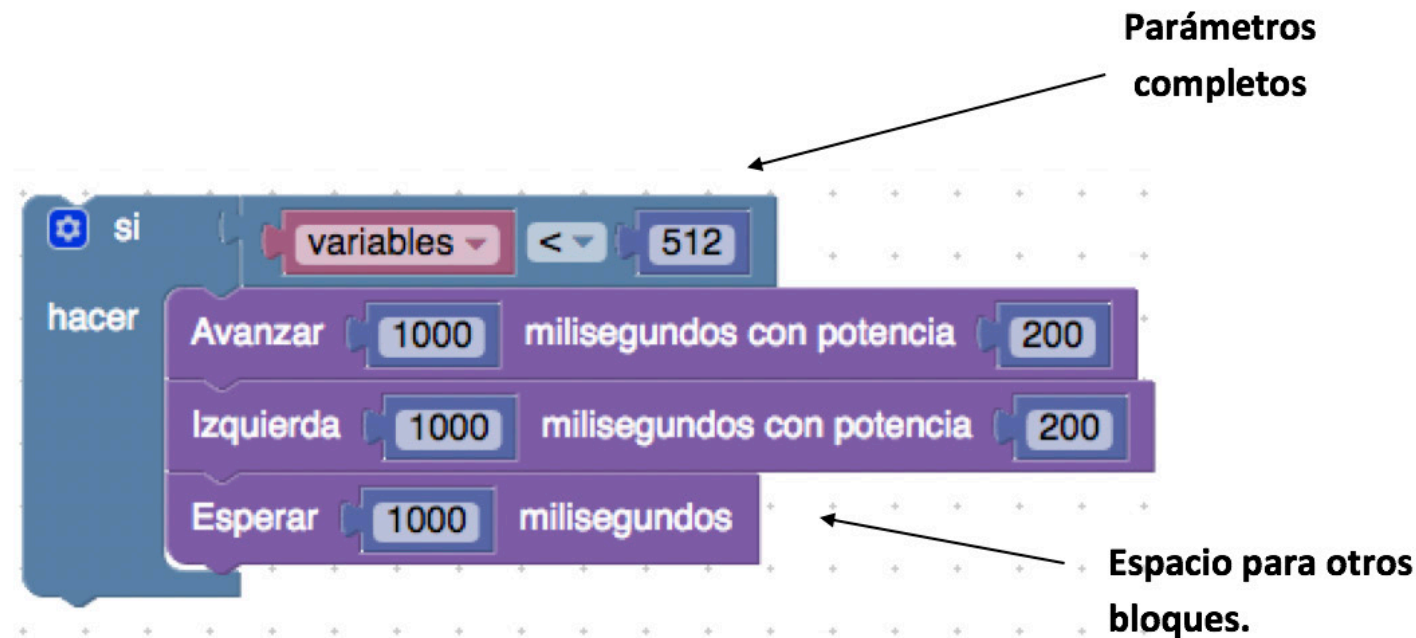
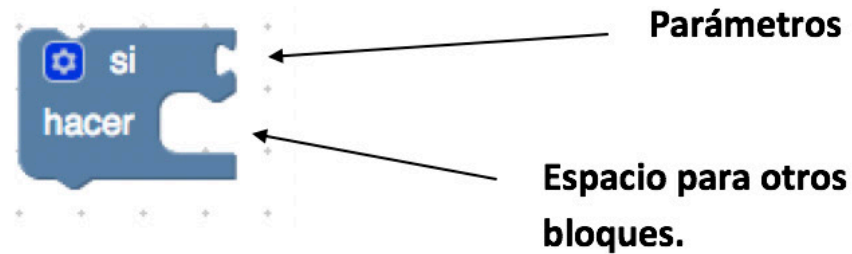


Parámetros



Parámetros
completos

Ejemplo de parámetros de los bloques y bloques elásticos.



Bloques Fundamentales

Bloques	Arduino	Libre
ComPacto	Iniciar comPacto	
Movimientos		
Motor		
Serial	SETUP	
Lógica		
Secuencias	LOOP	
Matemáticas		
Variables	Repetir	
	No Repetir	

Existen tres bloques fundamentales que deben estar presentes en todos los programas. Los tres se encuentran en el menú de **"Bloques" / "Compacto"**:

El primero es el bloque **"iniciar ComPacto"**. Es un bloque sin parámetros.

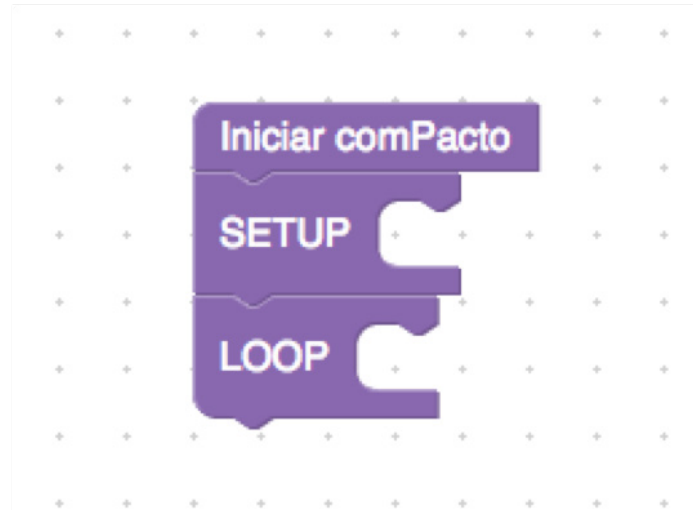
El segundo es el bloque **"Setup"** que es un bloque elástico. Dentro de él colocaremos todos bloques que se ejecutaran solo una vez y al inicio del programa.

El tercero es el bloque **"Loop"** que es un bloque elástico. Dentro del él colocaremos las acciones que realizara el robot. Una vez que culmine con la ultima acción, volverá al inicio para repetir indefinidamente todos los bloques.

Bloques Fundamentales

Los **tres bloques fundamentales** deben estar siempre en todos los programas. Unidos entre ellos uno debajo del otro y en orden.

El siguiente ejemplo es un programa funcional, correcto pero vacío.



Bloques Fundamentales

Realizaremos un programa sencillo como ejemplo.

El objetivo es que nuestro robot avance unos segundos, luego retroceda al mismo lugar y por último realice un giro a la izquierda.



Conversión de bloques a Arduino

¿Cómo pasar el
programa de bloques a
Arduino?

Bloques	Arduino	Libreria	XML
<pre>//Inicia Proyecto Compacto const int pinData = 8; const int pinClock = 4; const int pinLatch = 12; const int pinOE = 7; byte datoMultiplexor = 0; int pin_vell = 11; const int orden_dir1A = 2; const int orden_dir1B = 3; int pin_vel2 = 3; const int orden_dir2A = 4; const int orden_dir2B = 1;</pre>			

Una vez colocados los bloques se debe seleccionar del menú horizontal superior la opción **"Arduino"**. En esta pestaña, encontraremos el mismo programa que realizamos con bloques pero en **lenguaje por código**.

Este es el código que debo transportar al **Ide de "Arduino"**.

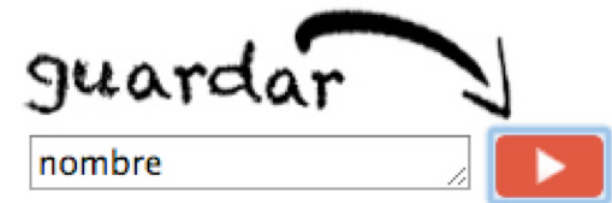
Conversión de bloques a Arduino.

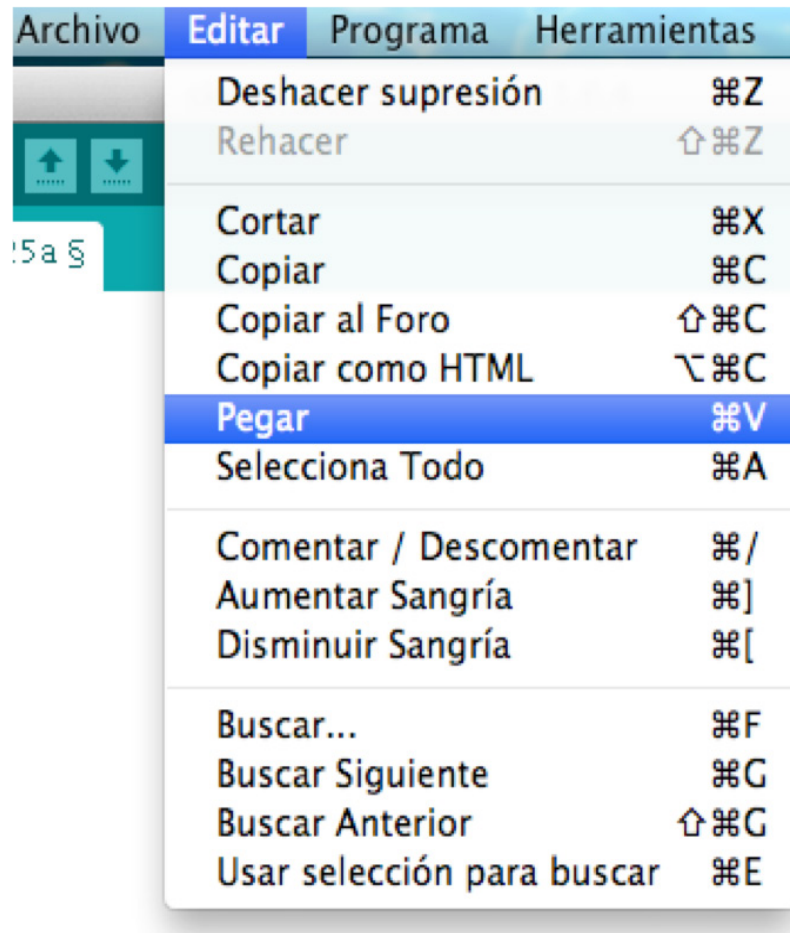
Como se ha explicado anteriormente existen dos formas de transportar el código al transportar al Ide de "Arduino".

1ra opción. Descargar el código en un archivo: pulsar el botón de guardar y se descargará un archivo que debo abrir con el Ide de "Arduino". Podemos agregar un nombre al archivo utilizando solamente letras sin espacios. El nombre se completará con el día, la hora minuto y segundo del instante en que se descarga el archivo.

El archivo se descarga en "Descargas" de la computadora. Al hacer doble click sobre el archivo se abrirá automáticamente con el programa Ide de "Arduino". También se puede abrir seleccionando el archivo descargado desde la barra de menú del Ide de "Arduino": **Archivo / abrir**.

Al abrir el Ide me mostrará un cartel pidiendo que se debe crear una carpeta con nombre: aceptamos esa necesidad.





Conversión de bloques a Arduino.

2da opción. Copiar el código completo generado en el Ide de **"ComPacto"** y pegar en el Ide de **"Arduino"** (asegúrese de copiar todo el código).

Para copiar el código: posicione el mouse en la parte superior del código presione el botón izquierdo y arrastre el mouse hasta el final sin soltar el botón. Una vez que pinte todo el código suelte el botón del mouse. Posteriormente sin despintar el código, presione sobre el botón derecho del mouse y seleccione del menú emergente la opción **"copiar"**.

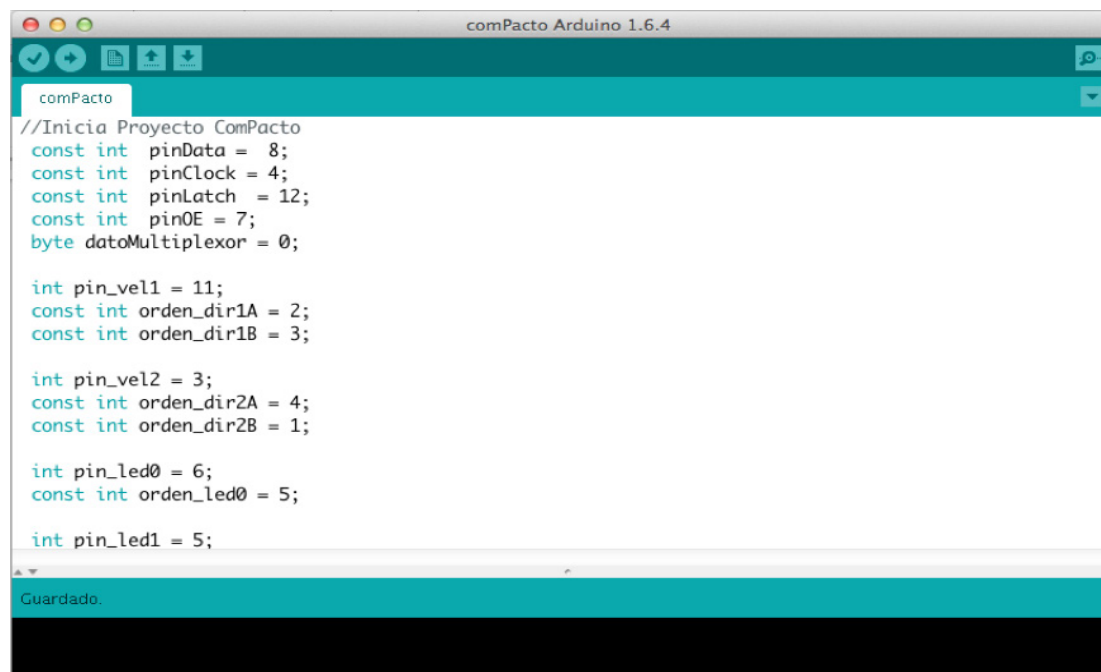
Una vez copiado, diríjase al **Ide de "Arduino"**. Tome una nueva hoja de trabajo y borre todo el texto que contenga y seleccione del menú desplegable la opción **pegar**.

¿Cómo cargar un programa al robot?

PASO N° 1.1

Código en el Ide de Arduino

Cargando programa al Robot.



```
comPacto Arduino 1.6.4
comPacto
//Inicia Proyecto ComPacto
const int pinData = 8;
const int pinClock = 4;
const int pinLatch = 12;
const int pinOE = 7;
byte datoMultiplexor = 0;

int pin_vel1 = 11;
const int orden_dir1A = 2;
const int orden_dir1B = 3;

int pin_vel2 = 3;
const int orden_dir2A = 4;
const int orden_dir2B = 1;

int pin_led0 = 6;
const int orden_led0 = 5;

int pin_led1 = 5;
```

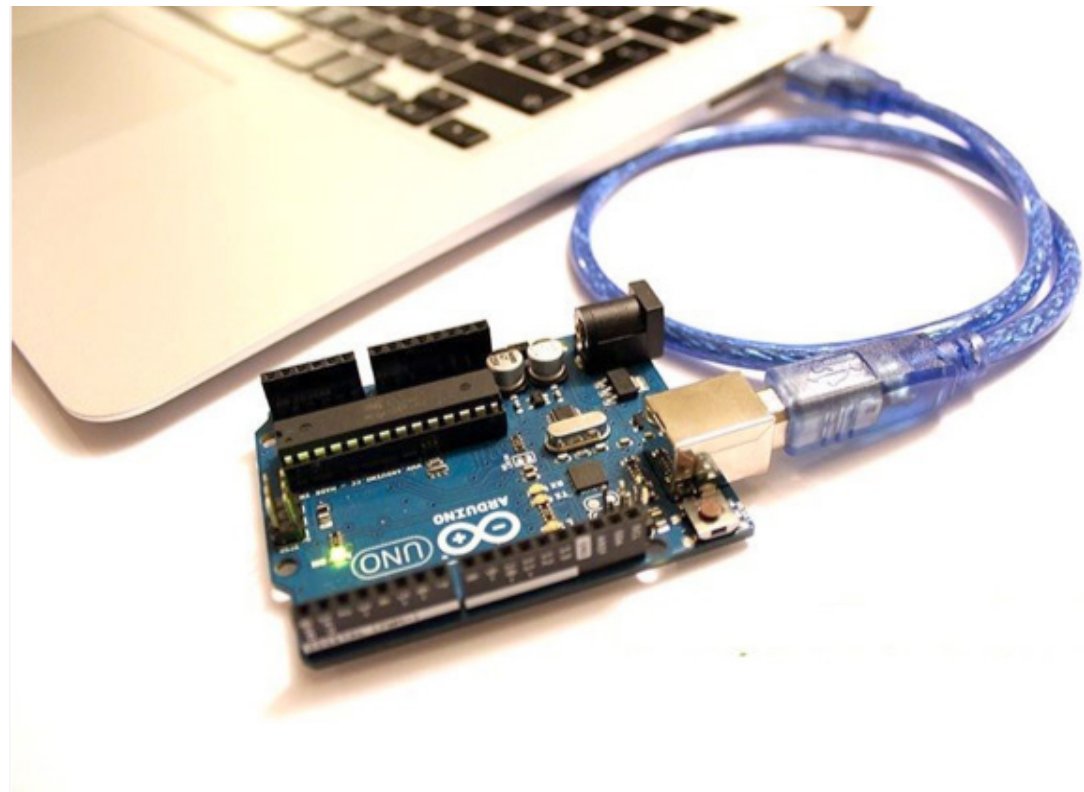
Guardado.

El primer paso es tener con seguridad el programa deseado en cuerpo del **Ide de "Arduino"**.

PASO N° 1.2 Conexión USB

Cargando programa al Robot.

El segundo paso es conectar el cable USB desde el controlador "Arduino" a la computadora.



PASO Nº 1.3 Selección de Placas

Cargando programa al Robot.

El próximo paso es seleccionar la **placa "Arduino"** que estamos utilizando. Es probable que solo se realice esto una vez. Existen varios modelos de placas **"Arduino"**, y cada una con sus propias características. En este paso le informaremos al **Ide de "Arduino"** que modelos estamos usando: **Arduino Uno / Genuino**. Vamos a la barra de menú horizontal, seleccionamos la opción **"Herramientas"** y luego dentro de la opción **"Placas"** buscamos la nuestra.

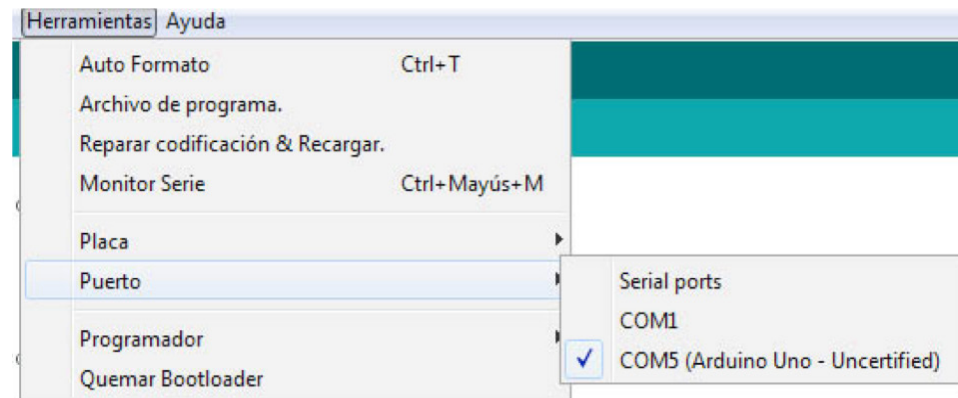


PASO N° 1.4 Selección de Puerto

Cargando programa al Robot.

El próximo paso es seleccionar en que puerto de la computadora hemos conectado la placa "**Arduino**". Esto se debe a que podemos tener conectados distintos **periféricos** o distintos "**Arduino**" simultáneamente, por ello debemos indicar cuál vamos a utilizar. También es probable que este paso lo tengamos que realizar solamente una vez o cada tanto.

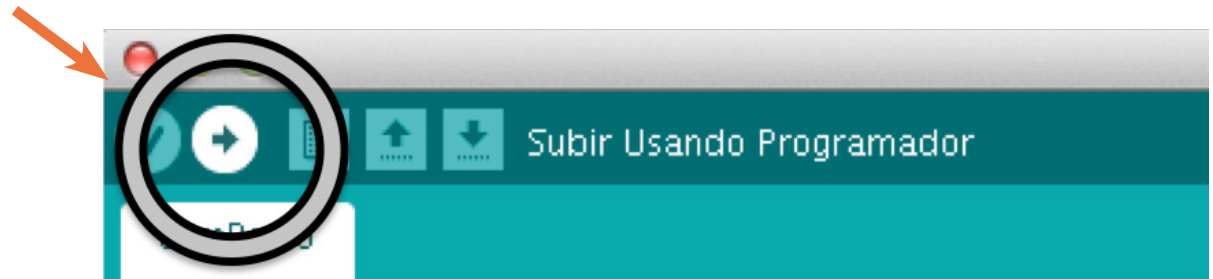
Vamos a la barra de menú horizontal, seleccionamos la opción "**Herramientas**" y luego dentro de la opción "**Puertos**" buscamos allí nuestra placa conectada. Si no aparece el nombre, tenemos que probar una por una hasta dar con la correcta. Los números de puerto lo asigna el sistema operativo de cada computadora.



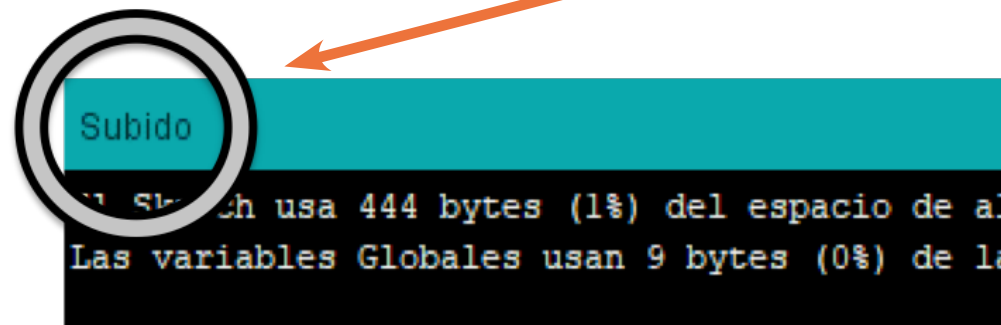
PASO N° 1.5 Puerto

Cargando programa al Robot.

El último paso es cargar el programa. Solamente seleccionamos el botón "Subir".



El proceso tardará unos segundos. Y finalmente en la barra superior de la terminal nos dará un mensaje de programa "Subido". El robot ejecutará el programa en unos instantes.

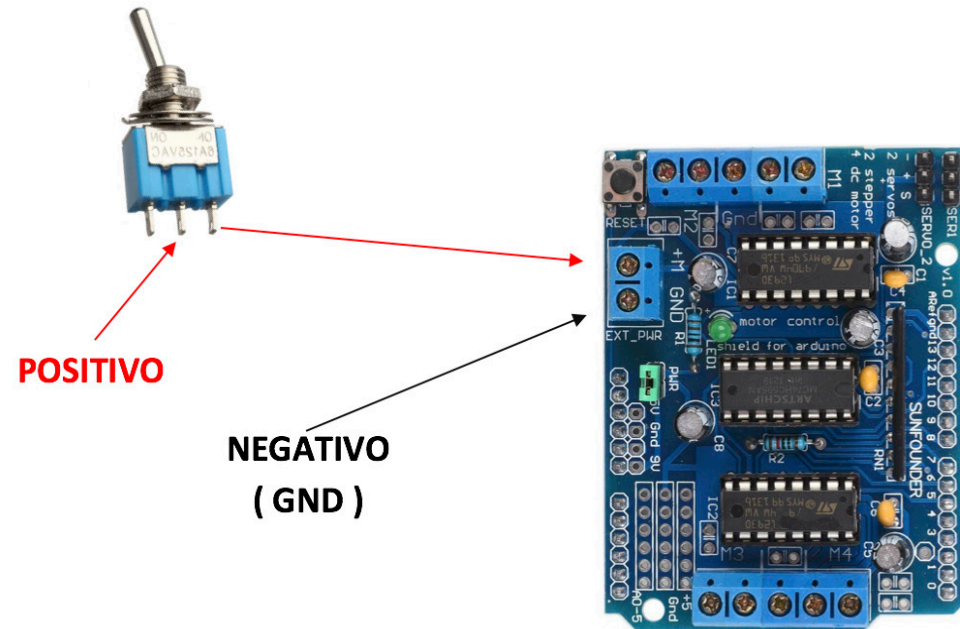


PASO N° 8.2

Conexiones de la batería

Conexiones de las Baterías.

Tener en cuenta que al realizar la conexión el robot arrancará. Si posee un programa anterior cargado se ejecutará y empezará a moverse. Para evitar esto es conveniente colocar un interruptor en el conector. Así se puede prender y apagar el robot desde allí.

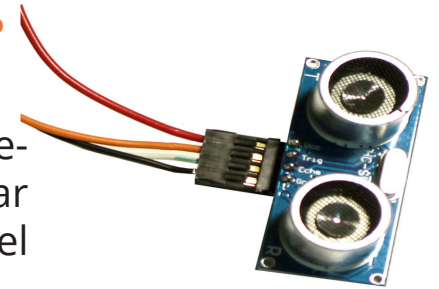


¿Cómo programar Sensores?

Sensor Ultrasónico

¿Qué es un sensor ultrasónico?

Un **sensor ultrasónico** es un detector de proximidad. Permiten calcular que distancia existe entre el sensor y el objeto más próximo frente a este.



El sensor consta de un emisor sonoro y un receptor. El sensor emite un sonido y mide el tiempo que tarda el sonido en golpear con un objeto y regresar. Ese tiempo permite calcular la distancia entre él y el objeto donde se reflejo el sonido. Estos objetos tiene un distancia máxima de uso, en el caso de nuestro programa posee una máximo valor de 100 cm.

Hay que tener presente que el sensor solo trabaja en forma lineal con un ángulo pequeño de trabajo, en otras palabras, el sensor ve solamente hacia delante y no hacia los costados. También solo detecta objetos grandes, como una pared, una caja, etc. Es menos probable que detecte objetos pequeños como la pata de una silla o de una mesa. Otro posible problema es si el sensor esta inclinado hacia el piso, es probable que entienda que el piso sea un objeto delante de él.

Sensor Ultrasónico

Bloque Ultrasónico

La conexión del sensor ya se ha explicado en este mismo tutorial.

El bloque que utilizaremos para el uso del sensor se llama “**Establecer Elemento con el valor de sensor Ultrasónico**”. Este bloque se encuentra en el menú “**ComPacto**”. El bloque sensor tiene un solo parámetro que es una variable. Por defecto se llama “**elemento**” pero puede cambiarse el nombre.



Una **variable** es en un espacio de la memoria donde se guardará el valor que el bloque devuelve:

El resultado de la medición de la distancia de 1 a 100 cm.

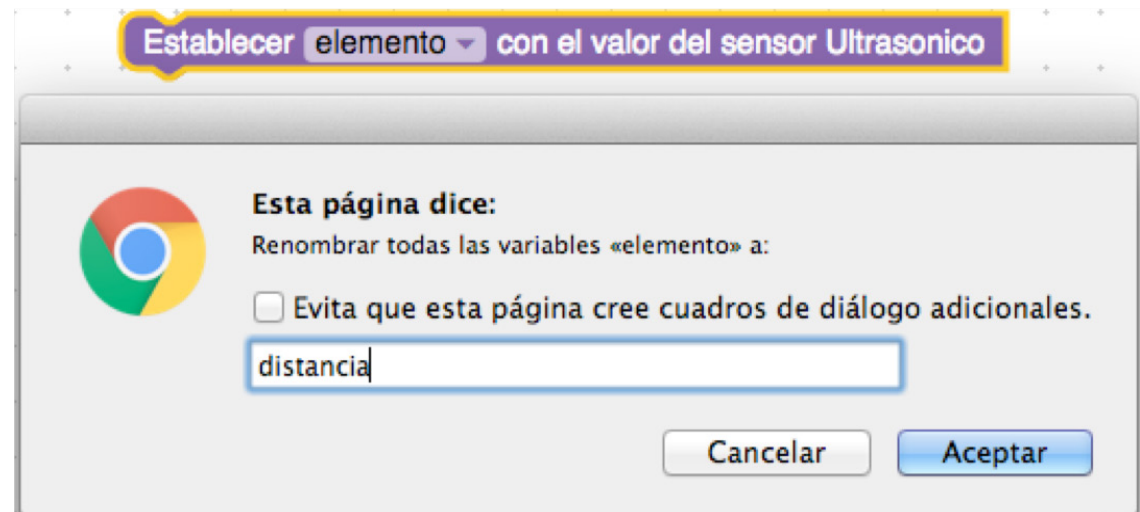
A ese espacio de memoria se le llama **variable** y puede **etiquetarse**:

Podemos llamarla con un nombre que nosotros inventemos. Por lo general, se le pone un nombre que describa el contenido del espacio. Puede pensarse como una caja donde guardo cosas y le pongo una etiqueta por fuera describiendo el contenido.

Sensor Ultrasónico

Nombre de una variable

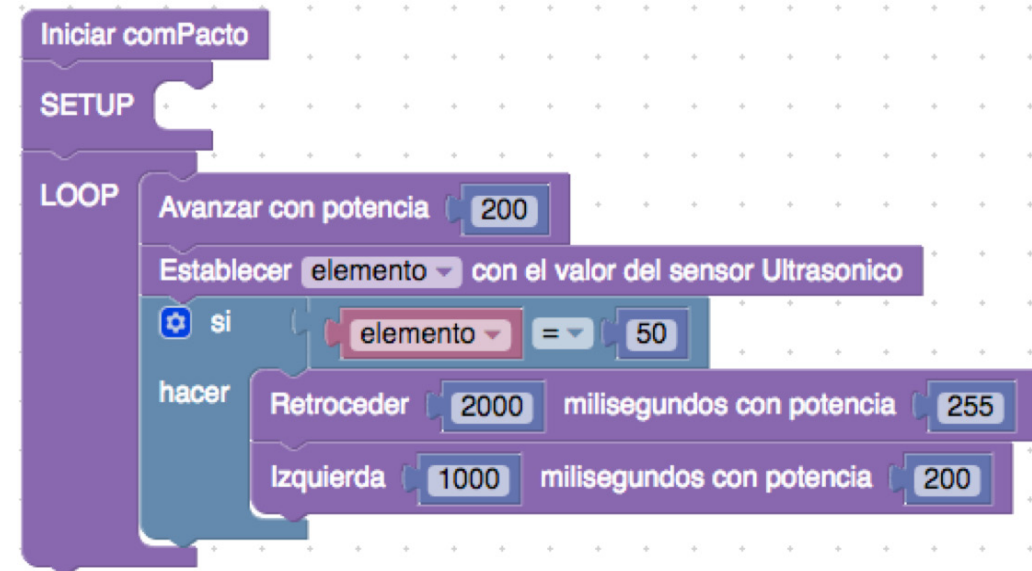
El bloque posee el nombre **"elemento"**, pero puede crearse uno nuevo, por ejemplo: **"distancia"**. Uno de los caminos para generar una nueva variable es haciendo **"click"** en el menú desplegable al lado del nombre **"elemento"** dentro del bloque. En este mismo, seleccionaremos la opción **"Renombrar"**. Por último asignaremos el nuevo nombre de la variable. El nombre puede estar formado por letras y números, **pero no puede empezar su nombre con un número**.



Sensor Ultrasónico

En resumen, el bloque **“Establecer elemento con el valor de sensor Ultrasónico”** recoge el valor que entrega el sensor ultrasónico y lo guarda en una **variable**. Por default lo guarda en una variable llamada **“elemento”** pero el bloque permite generar una nueva variable con el nombre que deseamos.

Lo importante es que con el valor de la medición realizada por el sensor, podemos tomar decisiones de comportamiento luego de realizar un análisis. Esa toma de decisiones se realiza con los **bloques de lógica**.



Sensor Ultrasónico

Bloque de lógica

En el menú vertical de **bloques "lógica"**, encontraremos los bloques necesarios para que el robot pueda tomar decisiones por sí mismo. Hasta ahora todos los ejemplos que hemos realizado han establecido los movimientos de nuestro robot linealmente: primero uno, luego otro y otro para finalmente volver a empezar.



Ahora podemos generar distintas opciones, por ejemplo utilizando el valor del sensor ultrasónico como disparador de distintos movimientos.

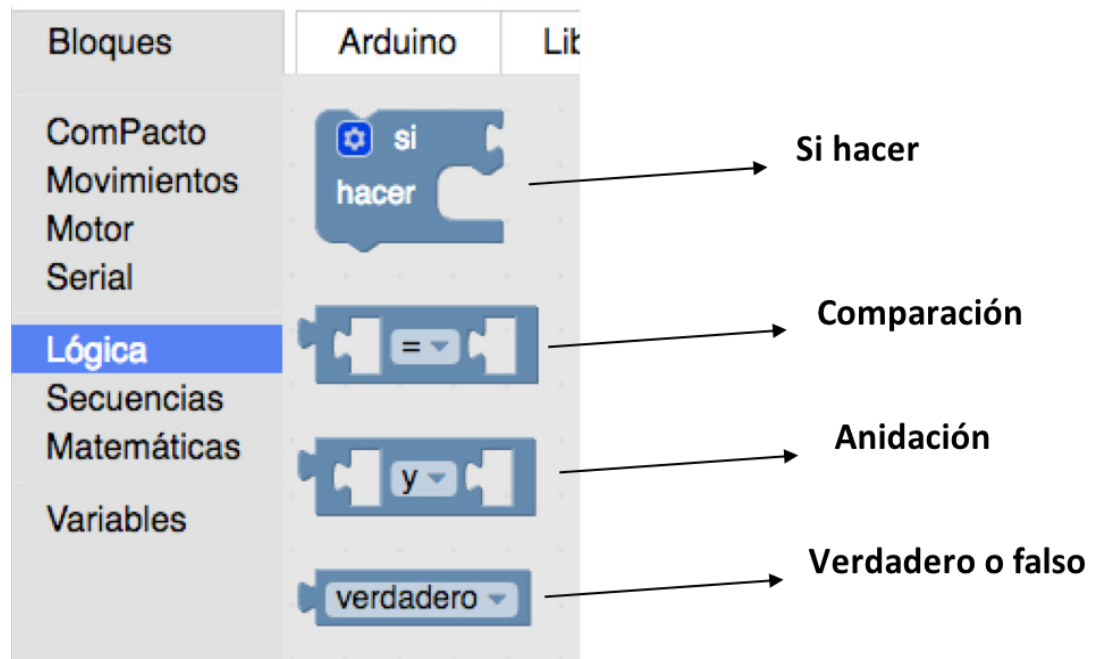
Por ejemplo: podemos hacer un programa para que el robot avance pero si y solo si el robot detecta un objeto delante de él, retroceda y gire a la izquierda. Con esta lógica no podríamos decir cuanto tiempo el robot avanzará, pero si sabemos que retrocederá y girará cuando detecte algún obstáculo.

Sensor Ultrasónico

Bloque de lógica

En el menú vertical de **bloques "lógica"**, encontraremos los bloques necesarios para que el robot pueda tomar decisiones por si mismo.

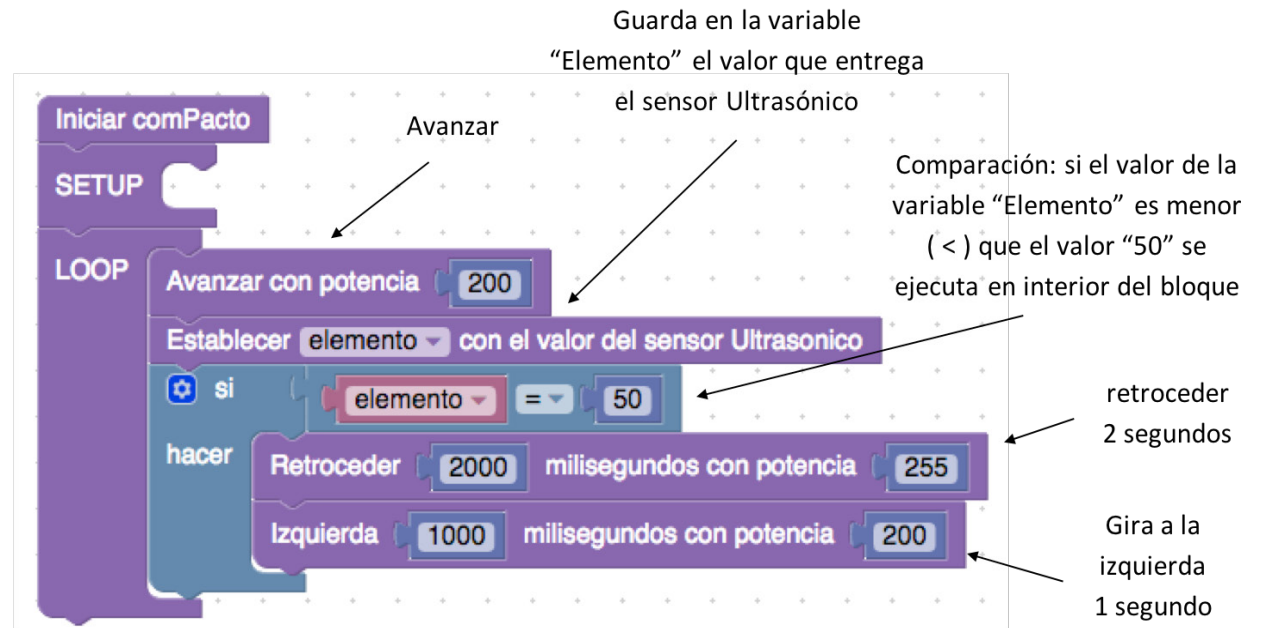
La utilización de cada bloque puede verse en las referencias de la web:
<http://www.roboticaalternativa.com/referenciasComPacto.html>
o en el material de descarga.



Sensor Ultrasónico

Ejemplo Sensor Ultrasónico

El robot avanza siempre, pero si y solo si el robot detecta un objeto delante de él: retrocede y gira a la izquierda.



Sensor Ultrasónico

Bloque de lógica

En el menú vertical de **bloques "lógica"**, encontraremos los bloques necesarios para que el robot pueda tomar decisiones por sí mismo. Hasta ahora todos los ejemplos que hemos realizado han establecido los movimientos de nuestro robot linealmente: primero uno, luego otro y otro para finalmente volver a empezar.



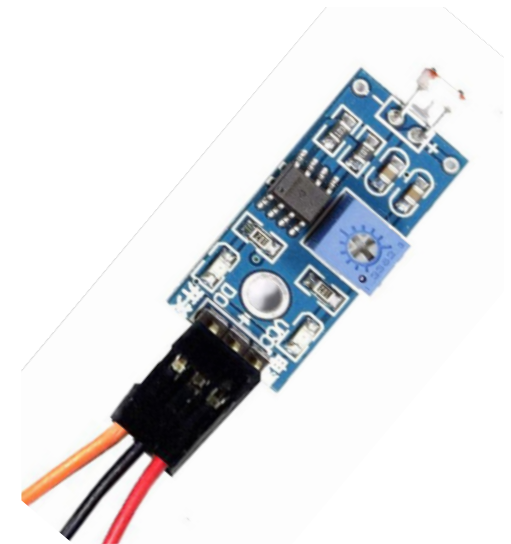
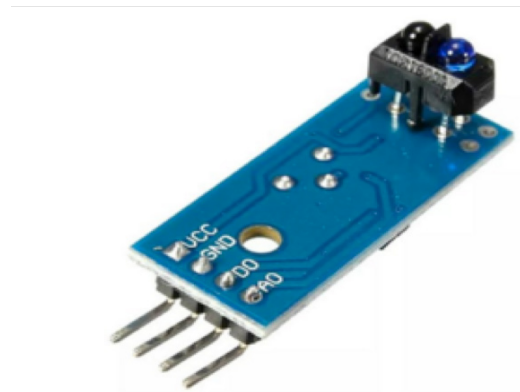
Ahora podemos generar distintas opciones, por ejemplo utilizando el valor del sensor ultrasónico como disparador de distintos movimientos.

Por ejemplo: podemos hacer un programa para que el robot avance pero si y solo si el robot detecta un objeto delante de él, retroceda y gire a la izquierda. Con esta lógica no podríamos decir cuanto tiempo el robot avanzará, pero si sabemos que retrocederá y girará cuando detecte algún obstáculo.

Sensores analógicos

Todos los sensores analógicos poseen la misma lógica de funcionamiento. Técnicamente, por lo general y a grandes rasgos, entregan un valor variables de 0 a 5 volt.

"ComPacto" admite aquellos sensores que respondan a esta características. Este proyecto propone la utilización de sensores LDR y infrarrojos. El rango de medición de un sensor analógico en "Arduino" se encuentra entre un valor de 0 a 1023.



Sensores Analógicos

¿Qué es un Sensor LDR?

Este sensor permite medir cantidad de luz, es una resistencia que varía con la llegada de fotones. A mayor luz será mayor la medida, que está en un rango de 0 a 1023. Cuidado: la conexión puede variar dependiendo el sensor, ya que existen distintos modelos, pero la lógica de conexión es la misma.

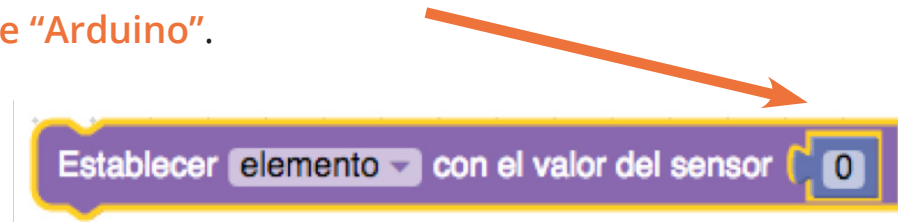
¿Qué es un Sensor Infrarrojo?

Este sensor posee un emisor de luz infrarroja y un receptor de luz infrarroja. Este módulo permite medir cuánta luz infrarroja rebota, de su propio emisor, sobre una superficie. A mayor luz de rebote, mayor será la medida, que está en un rango de 0 a 1023. Por lo general, si la superficie es clara o brillante rebota mayor cantidad de luz que una superficie oscura o rugosa. Cuidado: la conexión puede variar dependiendo el sensor, ya que existen distintos modelos, pero la lógica de conexión es la misma.

Sensores Analógicos

Bloque analógico

Los dos tipos de sensores propuestos utilizan los mismos bloques y lógica. Así que solo explicaremos como ejemplo de ambos uno: usaremos como ejemplo los LDR. El bloque que utilizaremos para el uso del sensor se llama **“Establecer Elemento con el valor de sensor”**. Este bloque se encuentra en el menú **“ComPacto”**. El bloque sensor tiene dos parámetros: una **variable** que se llama **“elemento”** pero puede cambiarse el nombre. El otro parámetro indica en que **pins** hemos conectado el sensor en la placa **shield de “Arduino”**.

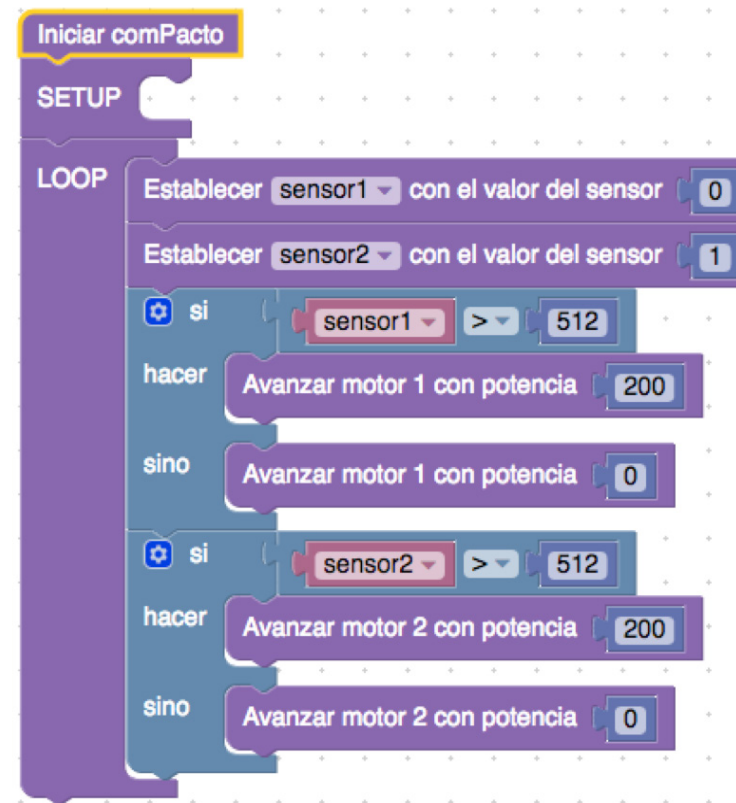


Una **variable** es un espacio de la memoria donde se guardará el valor que el bloque devuelve. A ese espacio de memoria se le llama variable y puede etiquetarse:

Podemos llamarla con un nombre que nosotros inventemos. Por lo general, se le pone un nombre que describa el contenido del espacio. Puede pensarse como una caja donde guardo cosas y le pongo una etiqueta por fuera describiendo el contenido.

Sensores Analógicos

En resumen, el bloque “Establecer elemento con el valor de sensor” recoge el valor que entrega el sensor analógico y lo guarda en una **variable**. Por default lo guarda en una variable llamada “**elemento**” pero el bloque permite generar una nueva variable con el nombre que deseamos.



Lo importante es que con el valor de la medición realizada por el sensor, podemos tomar decisiones de comportamiento luego de realizar un análisis. Esa toma de decisiones se realiza con los bloques de lógica.

Sensores Analógicos

Bloque de lógica

En el menú vertical de **bloques "lógica"**, encontraremos los bloques necesarios para que el robot pueda tomar decisiones por sí mismo. Hasta ahora todos los ejemplos que hemos realizado han establecido los movimientos de nuestro robot linealmente: primero uno, luego otro y otro para finalmente volver a empezar.



Ahora podemos generar distintas opciones, por ejemplo utilizando el valor del **sensor LDR** como disparador de distintos movimientos.

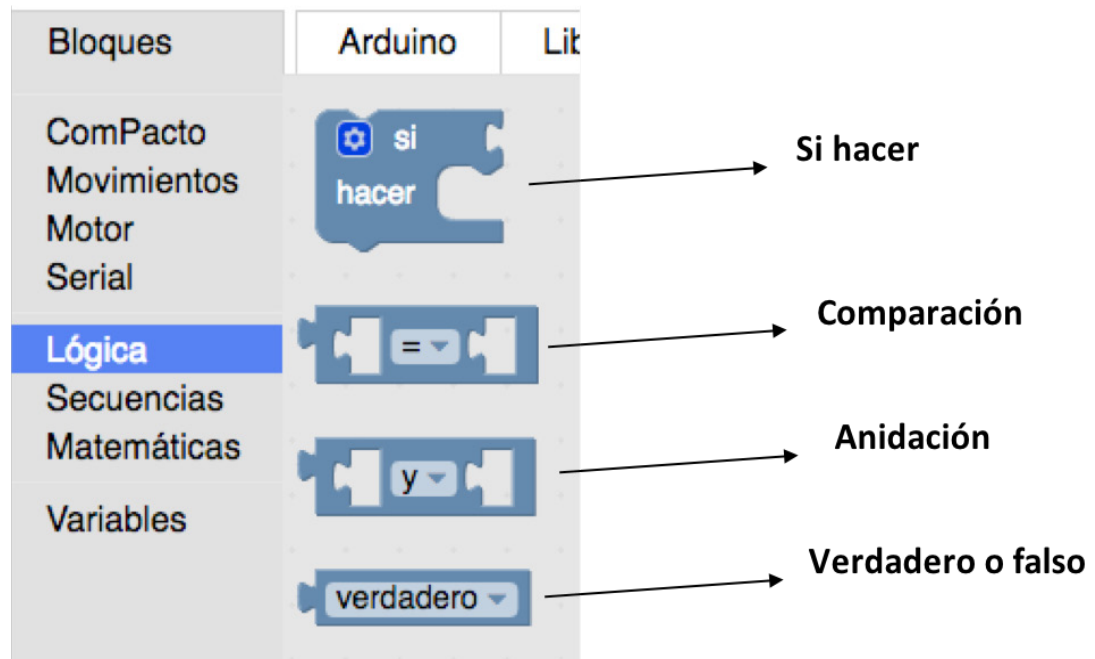
Por ejemplo: podemos hacer un robot que escape de la luz. Lo primero es armar un Robot con dos ruedas y colocar un sensor LDR cerca de cada rueda. Si logramos hacer un programa que cuando un sensor detecta luz encienda una rueda, el robot escapara de la luz.

Sensores Analógicos

Bloque de lógica

En el menú vertical de **bloques "lógica"**, encontraremos los bloques necesarios para que el robot pueda tomar decisiones por si mismo.

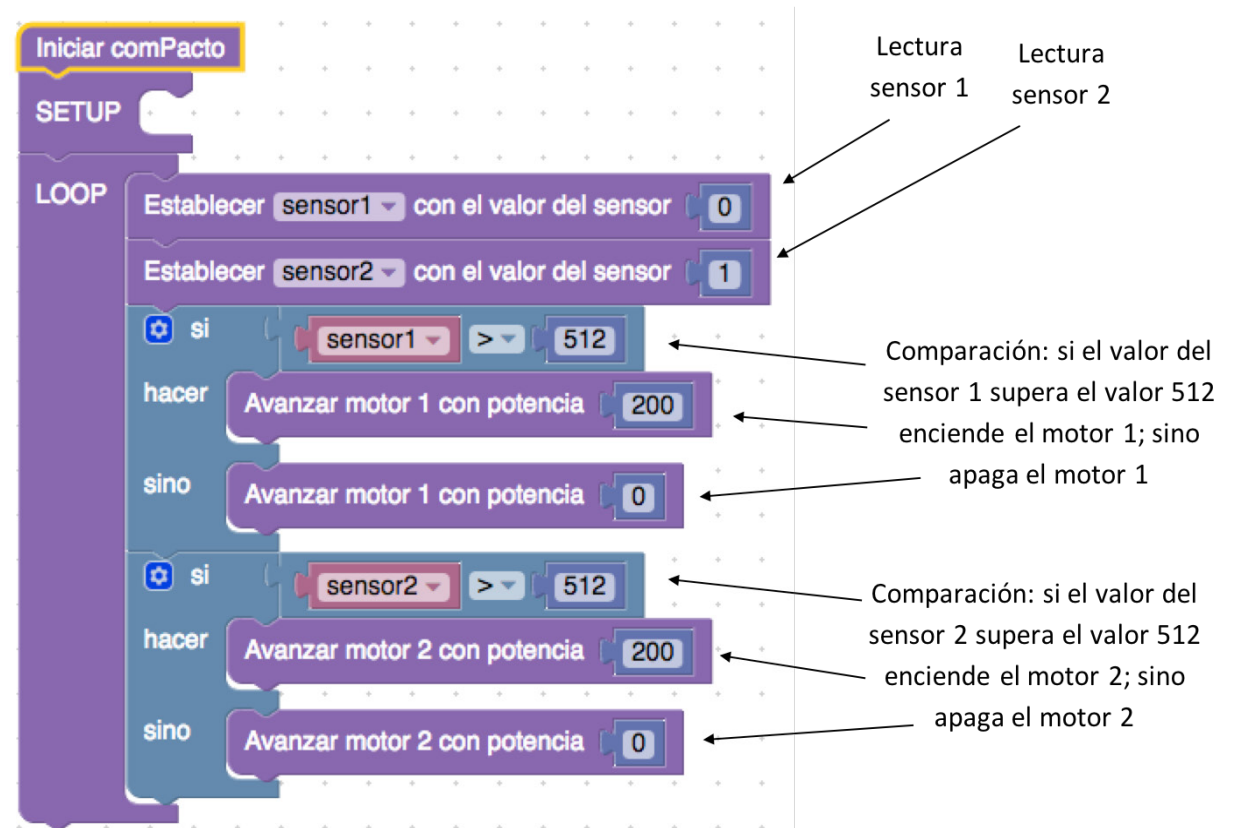
La utilización de cada bloque puede verse en las referencias de la web:
<http://www.roboticaalternativa.com/referenciasComPacto.html>
o en el material de descarga.



Sensores Analógicos

Ejemplo Sensor LDR

Hacer un Robot que escape de la luz.



ANEXO : Instalación de ARDUINO

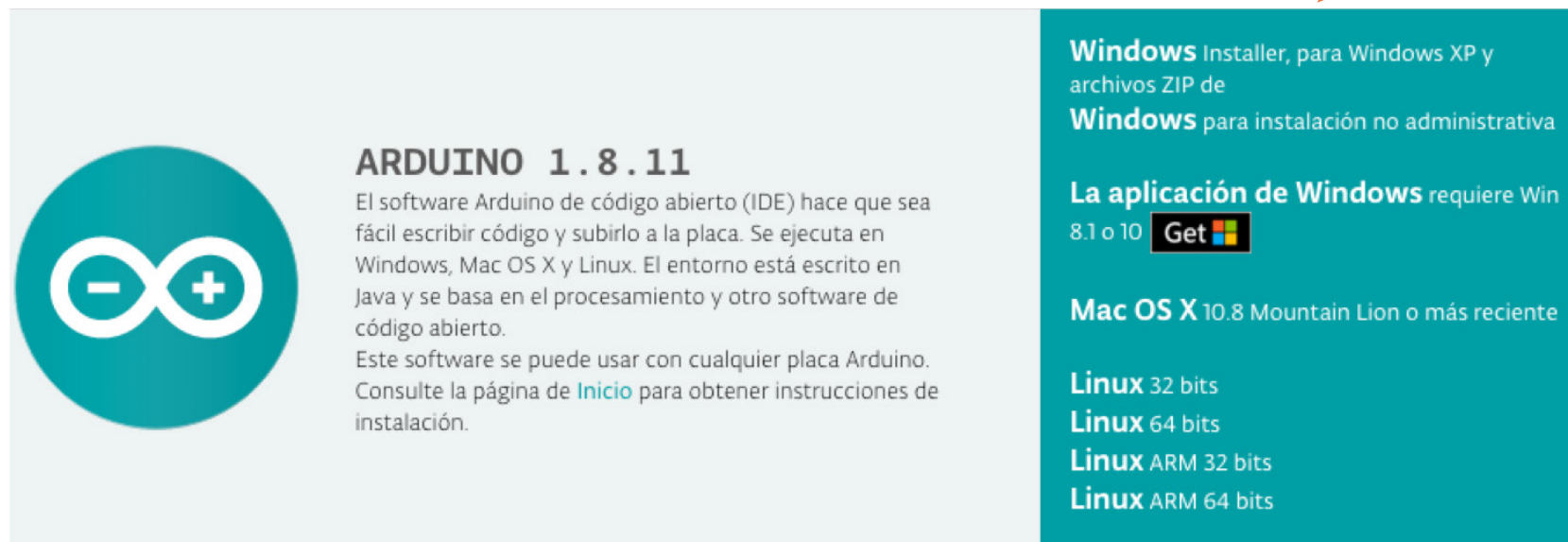
Instalación Arduino: <https://www.arduino.cc/>

El **Ide de "Arduino"** deberá descargarse de la página web de Arduino. En el menú superior, la pestaña **"SOFTWARE"**, hay que seleccionar la opción **"DESCARGAS"**.



Instalación Arduino: <https://www.arduino.cc/>

Posteriormente, se abrirá una nueva página. Deslizar el cursor hacia abajo y seleccionar el **Ide** a descargar que corresponda al sistema operativo de tu computadora.



ARDUINO 1.8.11

El software Arduino de código abierto (IDE) hace que sea fácil escribir código y subirlo a la placa. Se ejecuta en Windows, Mac OS X y Linux. El entorno está escrito en Java y se basa en el procesamiento y otro software de código abierto.

Este software se puede usar con cualquier placa Arduino. Consulte la página de [Inicio](#) para obtener instrucciones de instalación.

Windows Installer, para Windows XP y archivos ZIP de **Windows** para instalación no administrativa

La aplicación de **Windows** requiere Win 8.1 o 10 [Get](#)

Mac OS X 10.8 Mountain Lion o más reciente

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Instalación Arduino: <https://www.arduino.cc/>

La nueva página que se abrirá, visualiza la opción para contribuir monetariamente a los desarrollos de la plataforma Arduino. La contribución es opcional y el software a descargar seguirá siendo el mismo, independientemente de la decisión que tomes.



Instalación Arduino: <https://www.arduino.cc/>

Si descargas el software para **Windows**, tiene dos opciones. Un archivo **.exe** ejecutable o un archivo comprimido **.zip**. Si seleccionas el **.exe** ejecutable solo debes hacer “**doble click**” con el mouse para que se inicie el programa. Si seleccionas el **.zip** deberás descomprimir el archivo. Se generará un directorio dentro del cual se encontrará el **software de Arduino**.

Se recomienda utilizar el **.exe**, ya que al instalarse, también se instalarán los controladores (drivers). Esta es la guía oficial para Windows de Arduino: <https://www.arduino.cc/en/Guide/Windows>

Para descargar el software para **MAC OSX**, encontrarás un archivo comprimido en **.zip**. Descarga, descomprime y por último, arrastra el archivo del software a la carpeta de aplicaciones a modo de darle un orden. Esta es la guía oficial: <https://www.arduino.cc/en/Guide/MacOSX>

Instalación Arduino: <https://www.arduino.cc/>

Una vez instalado el software, hacer doble click sobre el programa. Si este fue instalado correctamente, se abrirá un ventana similar a la siguiente imagen:



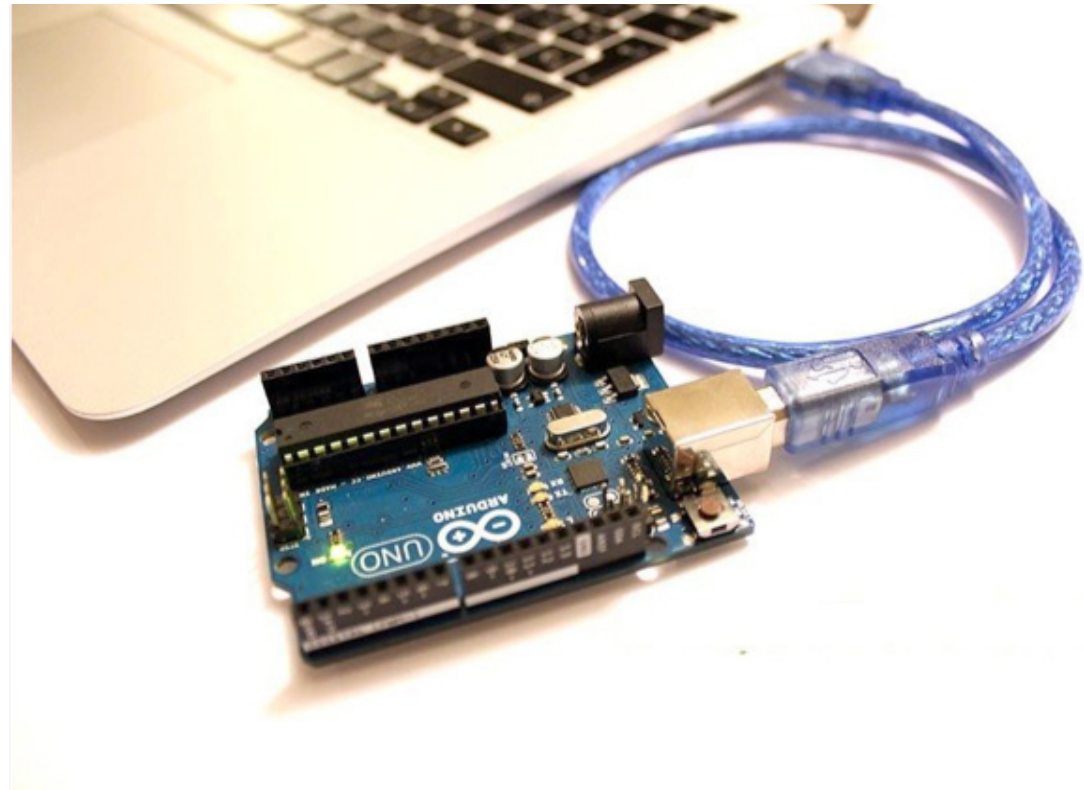
Test de Carga

PASO N° 1

Conexión USB

Test de Carga de un programa al "Arduino".

El primer paso es conectar el cable USB desde el controlador "Arduino" a la computadora.



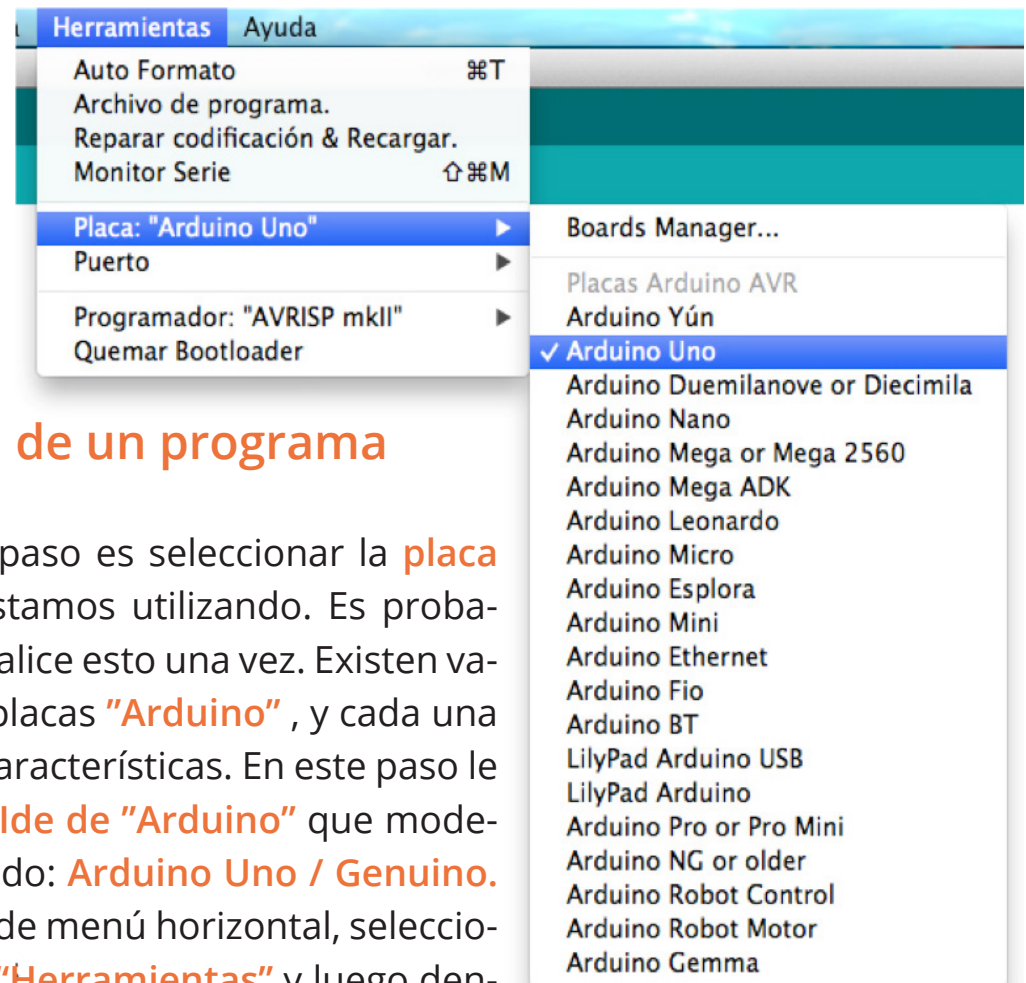
Test de Carga

PASO N° 2

Selección de Placas

Test de carga de un programa

El próximo paso es seleccionar la **placa "Arduino"** que estamos utilizando. Es probable que solo se realice esto una vez. Existen varios modelos de placas **"Arduino"**, y cada una con sus propias características. En este paso le informaremos al **Ide de "Arduino"** que modelos estamos usando: **Arduino Uno / Genuino**. Vamos a la barra de menú horizontal, seleccionamos la opción **"Herramientas"** y luego dentro de la opción **"Placas"** buscamos la nuestra.



Test de Carga

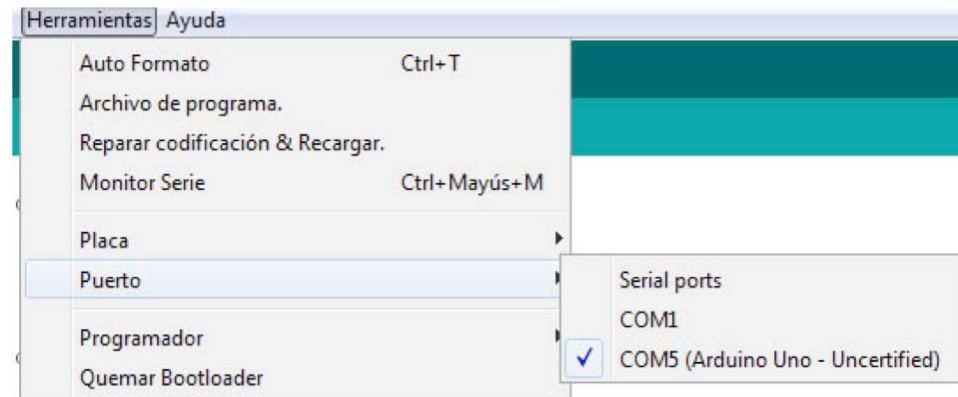
PASO N° 3

Selección de Puerto

Test de carga de un programa

El próximo paso es seleccionar en que puerto de la computadora hemos conectado la placa "**Arduino**". Esto se debe a que podemos tener conectados distintos **periféricos** o distintos "**Arduino**" simultáneamente, por ello debemos indicar cuál vamos a utilizar. También es probable que este paso lo tengamos que realizar solamente una vez o cada tanto.

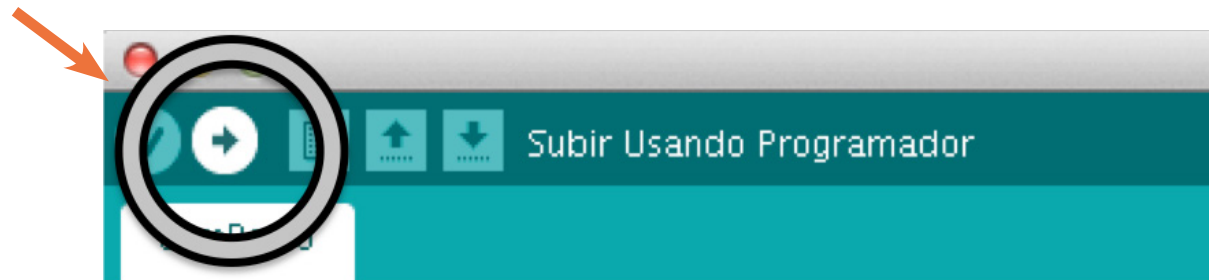
Vamos a la barra de menú horizontal, seleccionamos la opción "**Herramientas**" y luego dentro de la opción "**Puertos**" buscamos allí nuestra placa conectada. Si no aparece el nombre, tenemos que probar una por una hasta dar con la correcta. Los números de puerto lo asigna el sistema operativo de cada computadora.



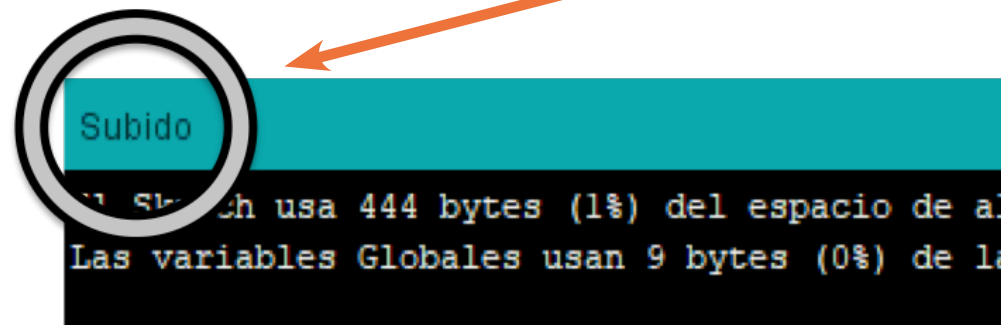
PASO N° 4 Puerto

Cargando programa al Arduino

El ultimo paso es cargar el programa. Solamente seleccionamos el botón "Subir".



El proceso tardara unos segundos. Y finalmente en la barra superior de la terminal nos dará un mensaje de programa "Subido". El robot ejecutará el programa en unos instantes.



arte

www.espacionixso.com.ar

info@espacionixso.com.ar

 [@espacionixsoarte](https://www.facebook.com/espacionixsoarte)

 [@espacionixso](https://www.instagram.com/espacionixso)



tecnología

 **espacionixso**
laboratorio de produccion artística