

Introducción a los Robots



Ley de Ohm

2

George Simon Ohm, formuló en 1827 la que se conoce como Ley de Ohm. Posiblemente una de las leyes fundamentales de la electrónica.

Primero definió matemáticamente las tres magnitudes físicas principales de la electrónica:

- Voltaje
- Intensidad
- Resistencia



Ley de Ohm

3



Ley de Ohm:

La Corriente en un circuito eléctrico varía de manera directamente proporcional a la Diferencia de Potencial (voltaje), e inversamente proporcional a la Resistencia.

$$I = \frac{V}{R}$$

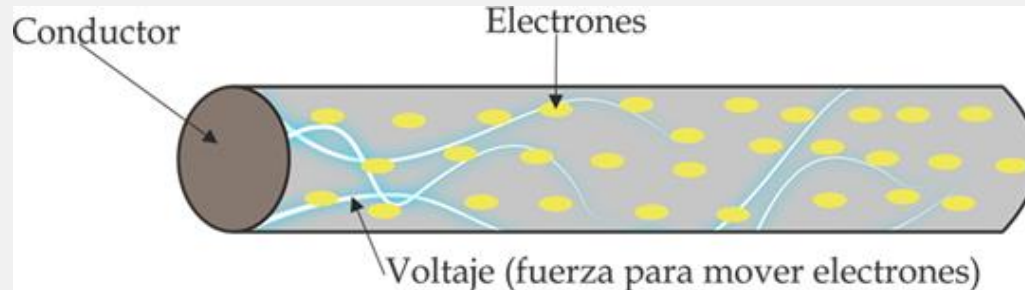


Conductividad:

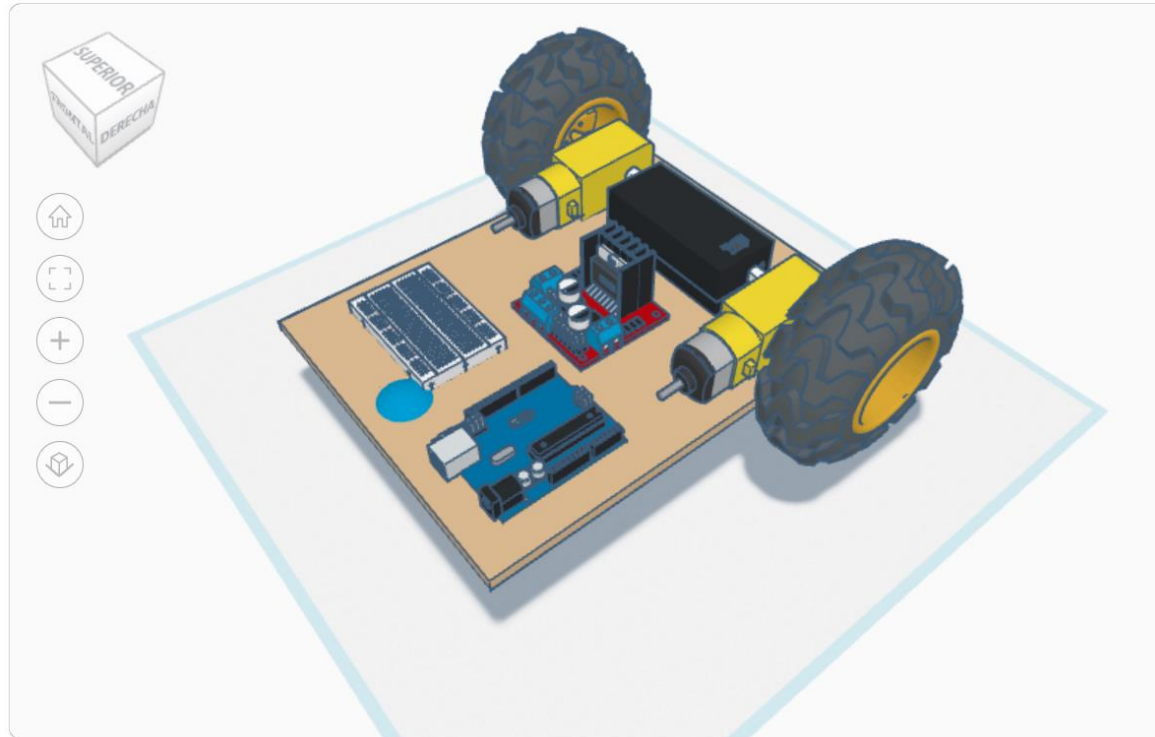
Para poder crear un camino entre dos voltajes, se utiliza un conductor.

Los conductores más utilizados son los cables.

El mejor conductor es el metal, y de éstos el mejor es el cobre.



Ya tengo mi diseño. ¿Y luego?



Anatomía



Proceso de construcción y elementos de un robot

Aunque todos los robots varían en su complejidad, todos se van armando tomando la siguiente secuencia:

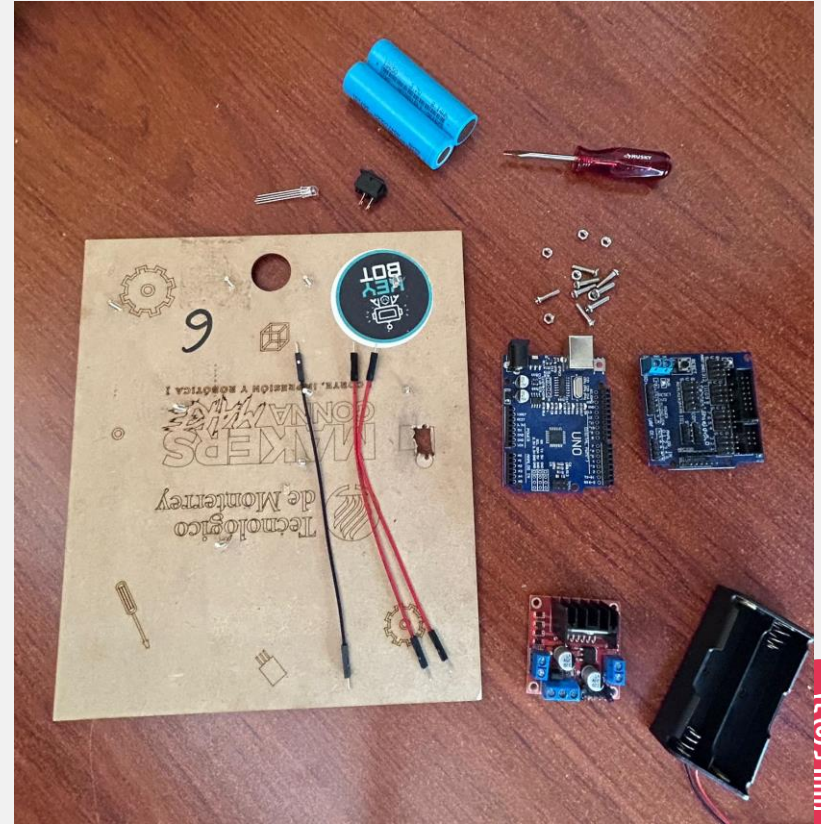
- A) S3 - Diseño Chasis y componentes principales
- B) S4- Instalación y cableado Sistema eléctrico
- C) S5- Programación del Sistema de control
- D) S6- Cableado, instalación de Actuadores y variadores
- E) S7- Cableado y programación de Sensores
- F) S8- Configuración y programación del sistema de Comunicación
- G) S9 -Pruebas

Armado



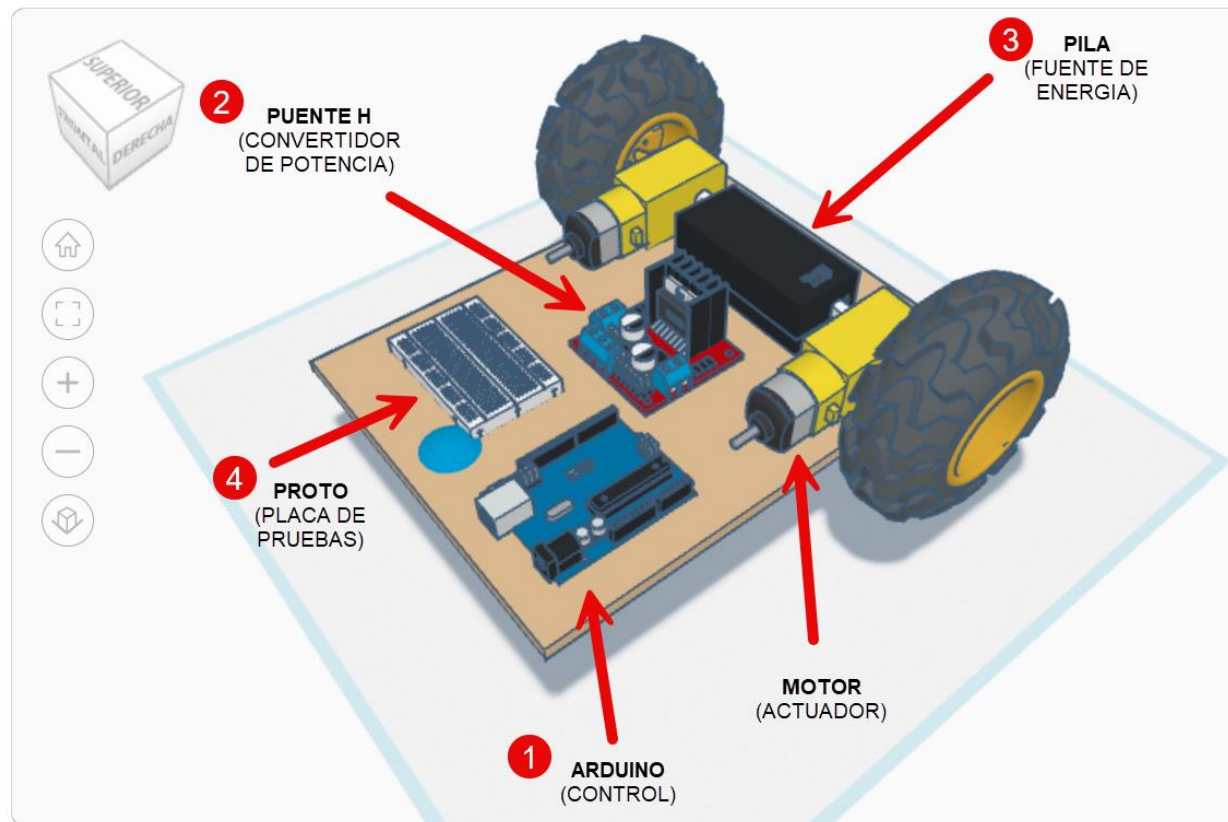
Material

- 1 Base MDF
- 1 Arduino / 1 Expansión
- 1 puente H L298N
- 9 tornillos M3
- 4 tuercas M3
- 1 Porta pilas
- Pilas 4volts
- 2 cables rojos (VCC)
- 1 cable negro (GND)



Armando nuestro BOT

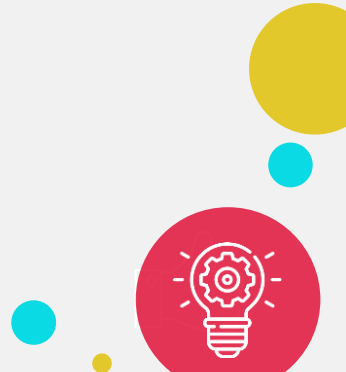
En esta sesión
trabajaremos en
el ensamble de
nuestro chasis
BOT



Chasis / Base

Primero la base

Con un
desarmador, coloca
los 9 tornillos M3
en la base.



1. Control / Arduino

¿Qué es Arduino?

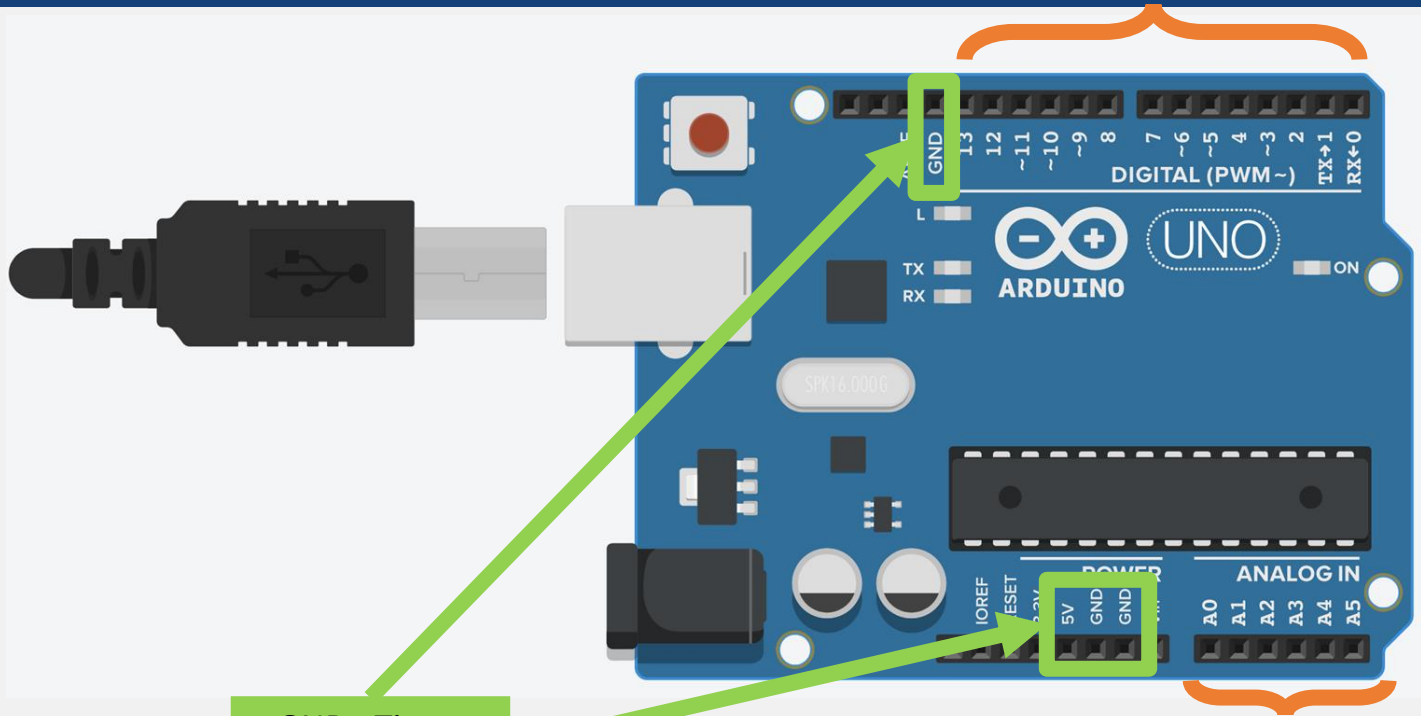
- Arduino es una plataforma de control de código abierto inventada en 2005 para hacer proyectos de forma sencilla.
- En FIRST nuestro sistema de control se llama **roboRIO**



Placa Arduino UNO

Pines Digitales

16



GND= Tierra
5V= voltaje

Pines Analógicos



Para que usar una tarjeta de expansión

- En Arduino, las tarjetas de expansión ayudan a crear una disposición más sencilla de los elementos de Arduino, en este caso nosotros utilizaremos el Sensor Shield, el cual se utiliza mucho en robótica.

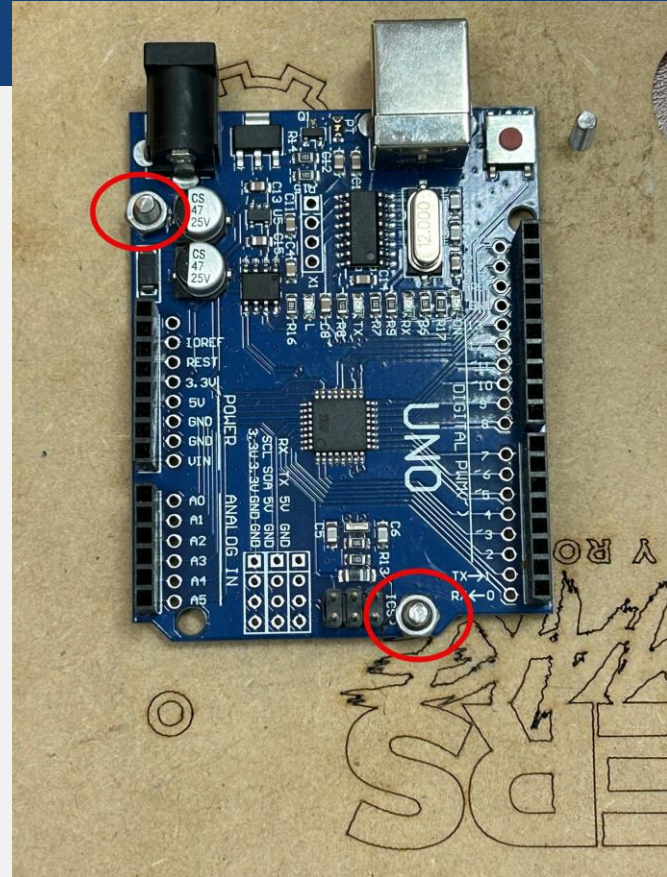
CAUTION: Colocar esta tarjeta requiere de cuidado para no enchuecar los pines, presta atención.



Fija el arduino

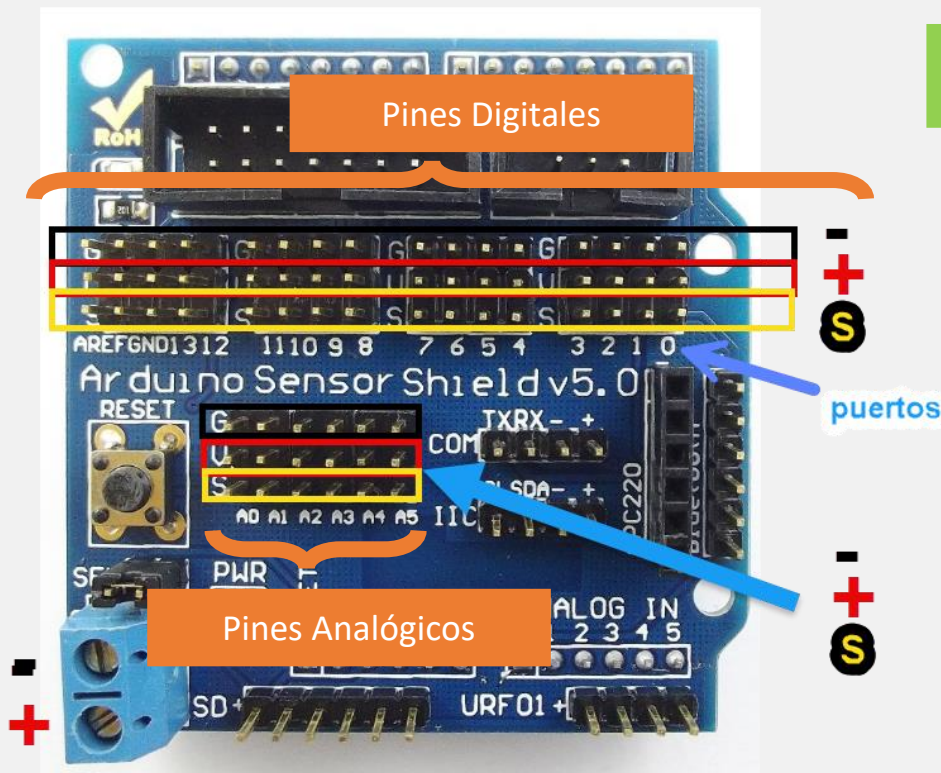
Una vez que haz colocado los tornillos M3, ahora coloca tu placa Arduino y sujétala con dos tuercas.

Al ser componentes delicados, aprieta las tuercas con tus manos.



TARJETA DE EXPANSIÓN – Sensor Shield V5.0

19

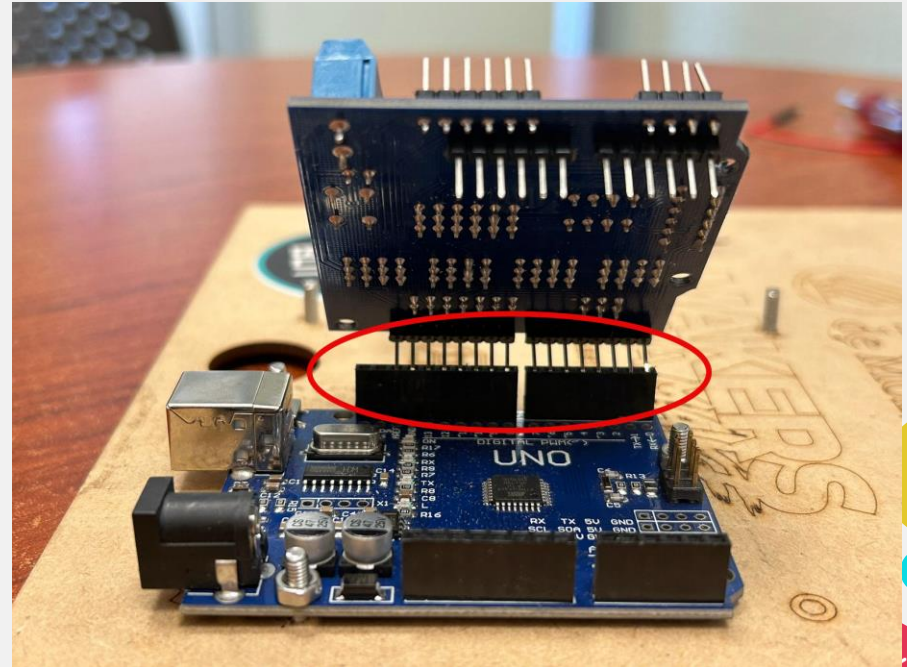


GND= Tierra
5V= voltaje



Coloca la Tarjeta de expansión

Recuerda que este paso requiere especial cuidado. Se recomienda primero presentarla de lado y bajarla con cuidado, como se muestra en la imagen.

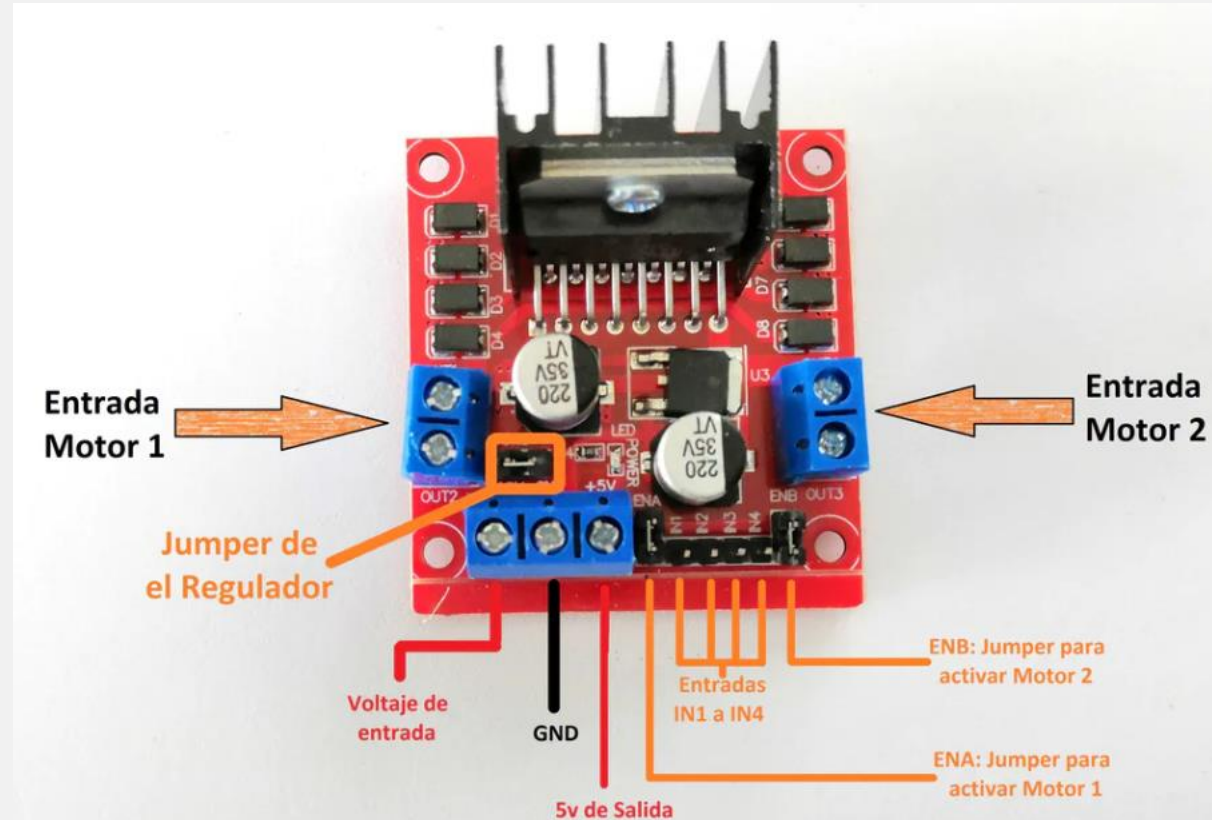


2. Puente H / L298N

¿Qué es el puente H?

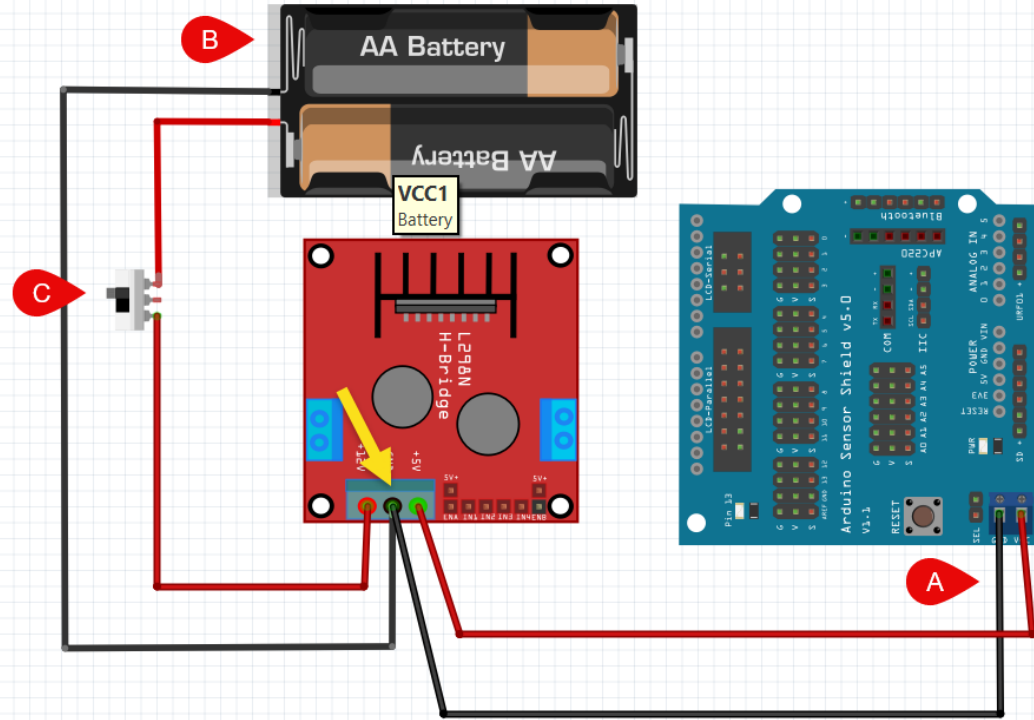
22

- Este componente permite controlar los motores, en FIRST los llamamos “Talons”



Conexiones

A partir del puente H y el sistema de control, vamos a cablear el siguiente circuito. En el siguiente orden.



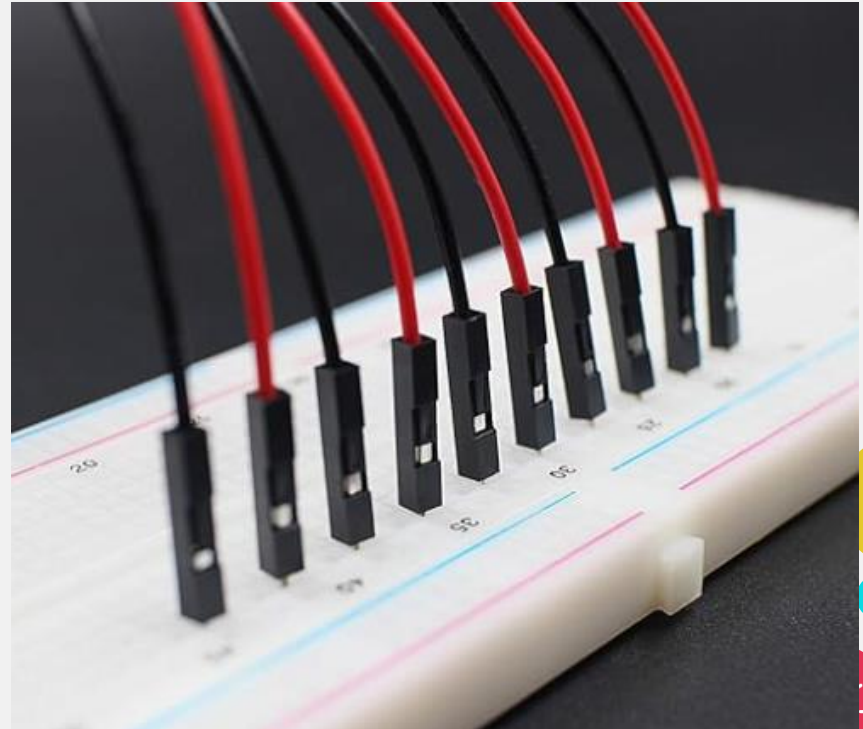
Se recomienda no fijar el puente H hasta probar todo tu circuito completo.

Buenas prácticas

Nota que los cables que usaremos son solo rojos y negros:

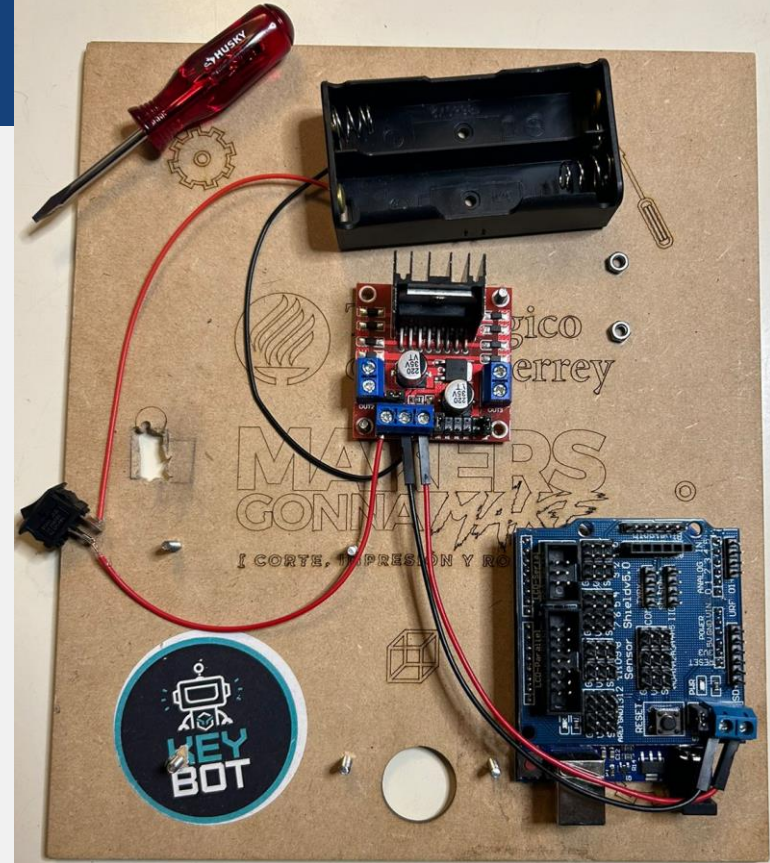
GND – negativo
(viene de GROUND o Tierra)

VCC – positivo
(Viene de Voltaje de corriente continua)



Cablea tu circuito

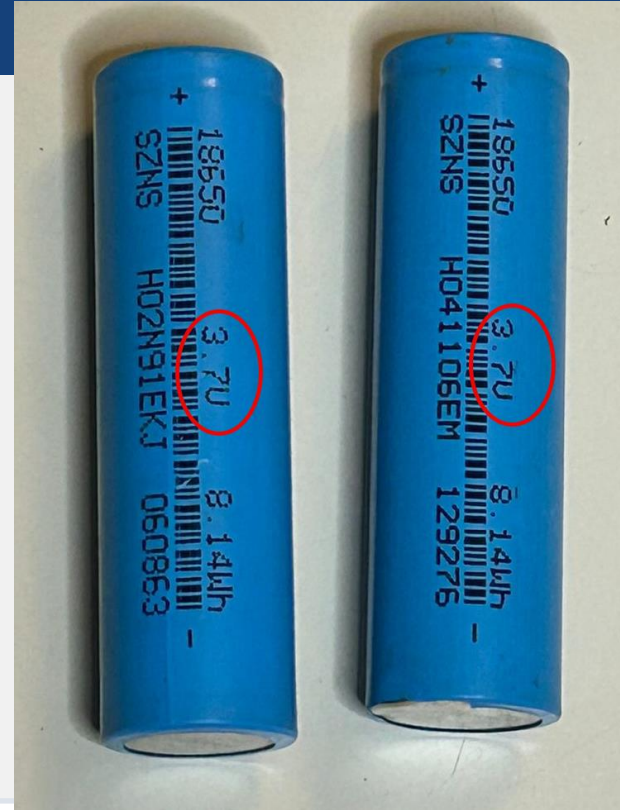
Utiliza un desarmador para colocar los cables al puente H y a la tarjeta de expansión, recuerda colocar correctamente **positivo** y **negativo**.



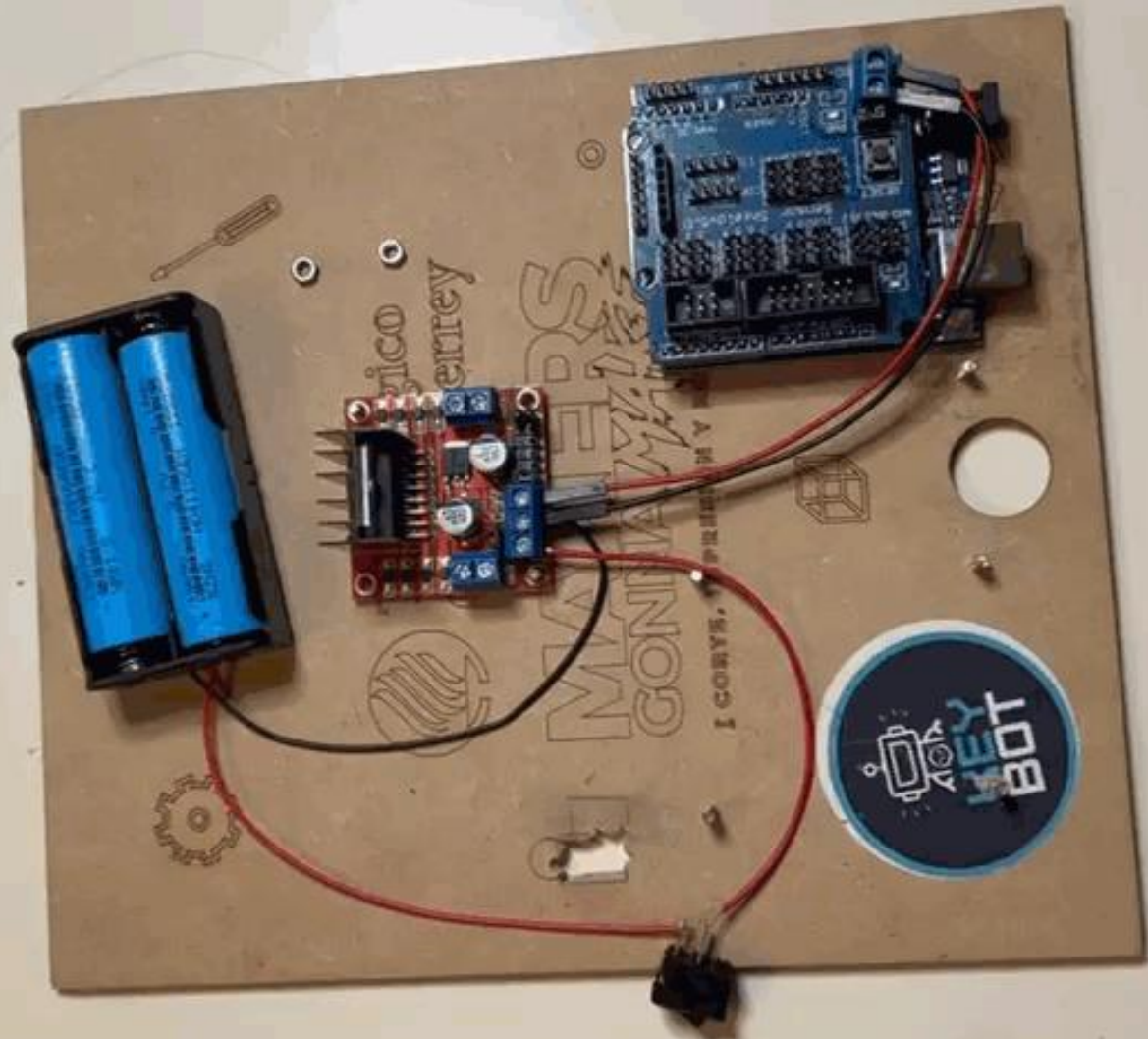
3. Pila 3.7v

Probar tu circuito

- Antes de fijar el puente H, el portapilas y el switch a la tabla, vamos a probar tu circuito con las pilas. Las pilas tienen polaridad por lo que debemos colocarlas de forma adecuada + y - .



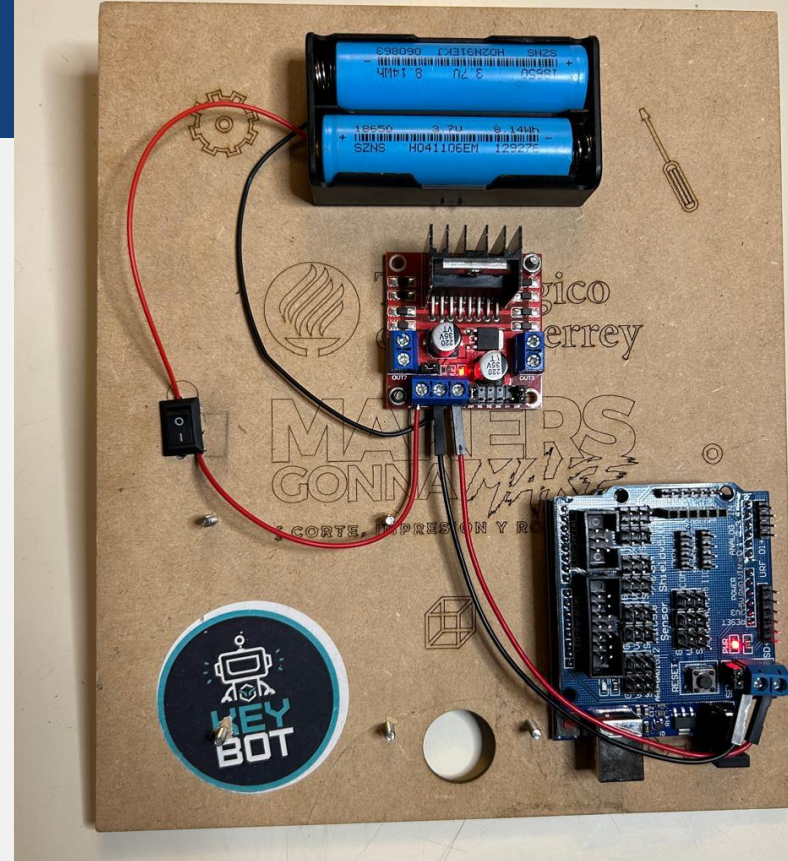
Se recomienda antes de colocar las pilas revisar que tengan carga.



Fijando los componentes

Ahora sí. Si tu circuito prende correctamente, es momento de fijarlo en tu tabla.

- Coloca las dos tuercas en tu puente H
- Fija el portapilas
- Fija el switch



Tu instructor pasará a revisar que todo este perfecto.

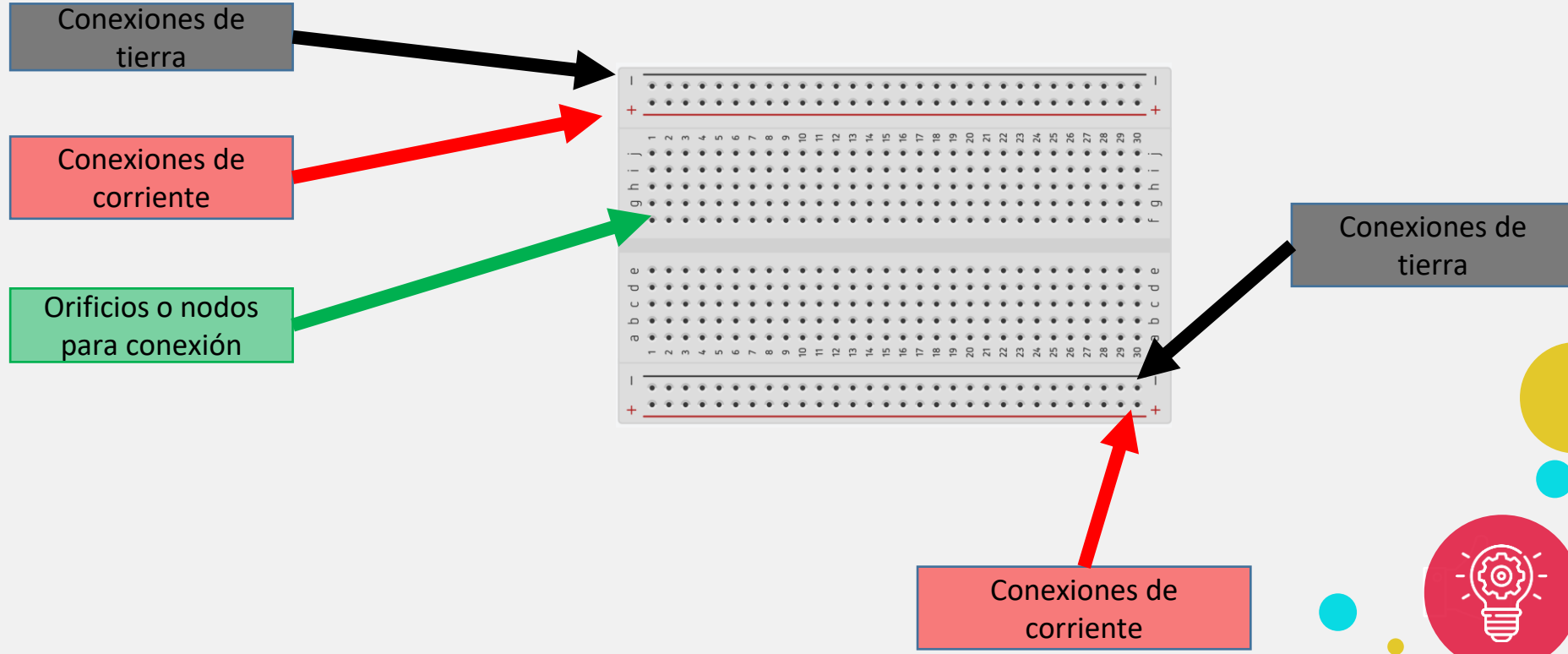
4. Breadboard

¿Qué es un breadboard / protoboard?

- El "protoboard" o "breadboard" es un tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares.

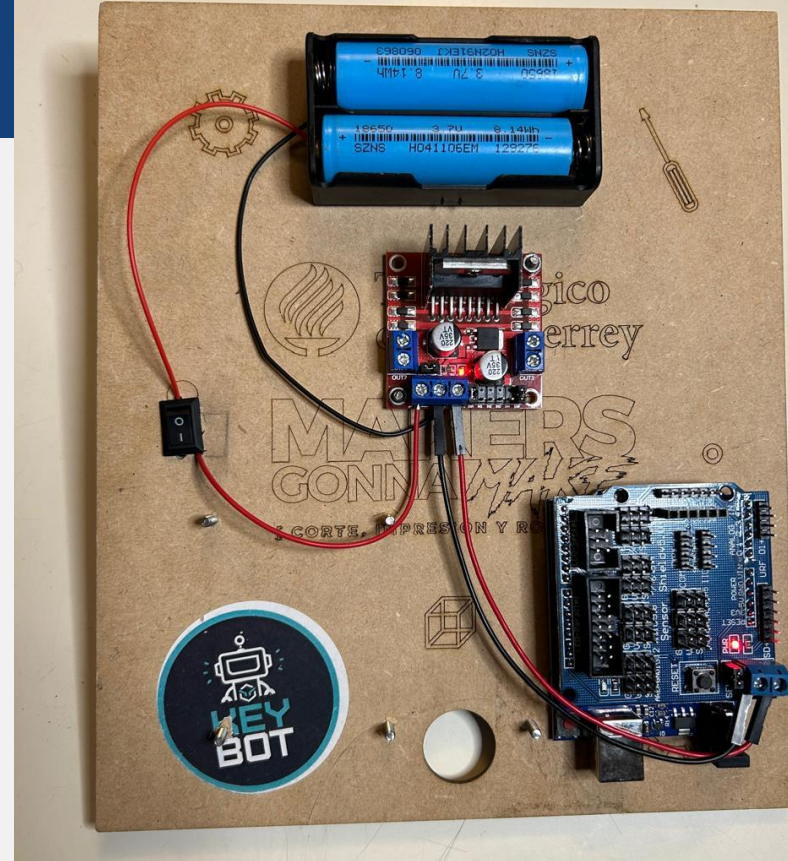


Protoboard o Breadboard



Colocando la breadboard

Coloca la protoboard junto al Arduino tal como se muestra en la foto.



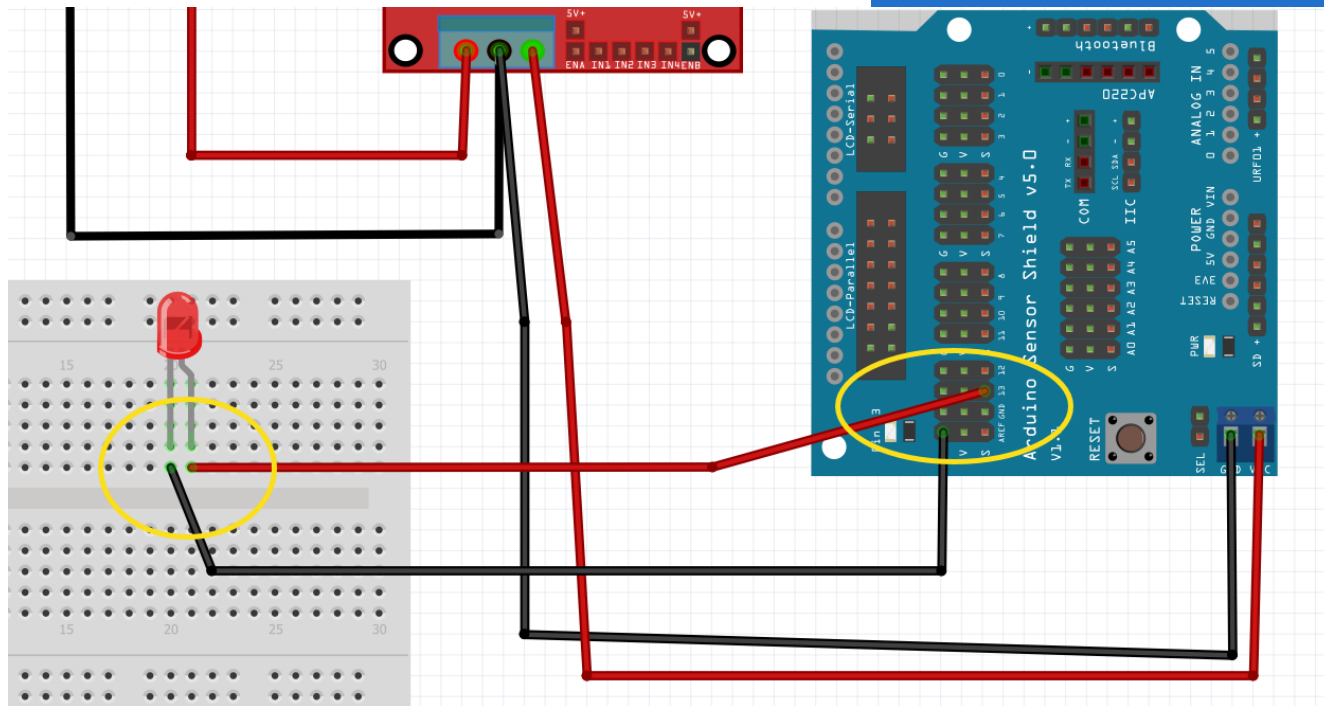
La breadboard entra a presión

iPractica!

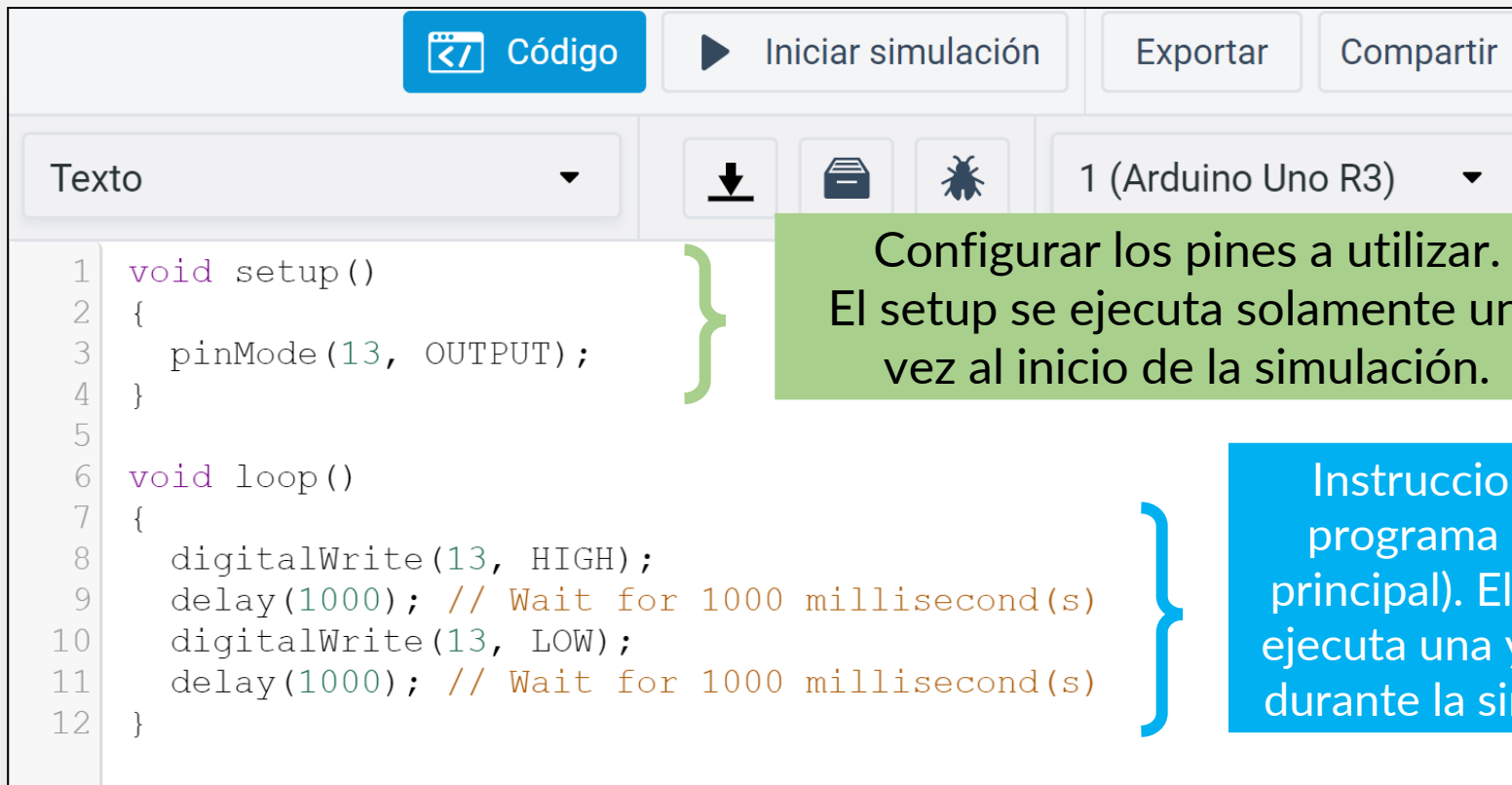
Vamos a realizar algunas pruebas con tu circuito



Circuito de prueba



Estructura del código



The screenshot shows the Arduino IDE interface. At the top, there are buttons for 'Código' (with a code icon), 'Iniciar simulación' (with a play icon), 'Exportar', and 'Compartir'. Below these is a toolbar with a 'Texto' dropdown, a download icon, a file icon, a bug icon, and a board selector set to '1 (Arduino Uno R3)'. The main area contains the following code:

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

Two callouts explain the code structure:

- A green callout points to the `setup()` function, stating: "Configurar los pines a utilizar. El setup se ejecuta solamente una vez al inicio de la simulación."
- A blue callout points to the `loop()` function, stating: "Instrucciones del programa (código principal). El loop() se ejecuta una y otra vez durante la simulación."

Comandos básicos de Arduino

`pinMode()`

- Configura el pin especificado para que se comporte como INPUT o OUTPUT.

`digitalWrite()`

- Escribe un valor HIGH o LOW en un pin digital.
- HIGH = 1 u ON
- LOW = 0 u OFF

`delay()`

- Pausa el programa por la cantidad de tiempo (en milisegundos) especificado como parámetro. (Hay 1000 milisegundos en un segundo).

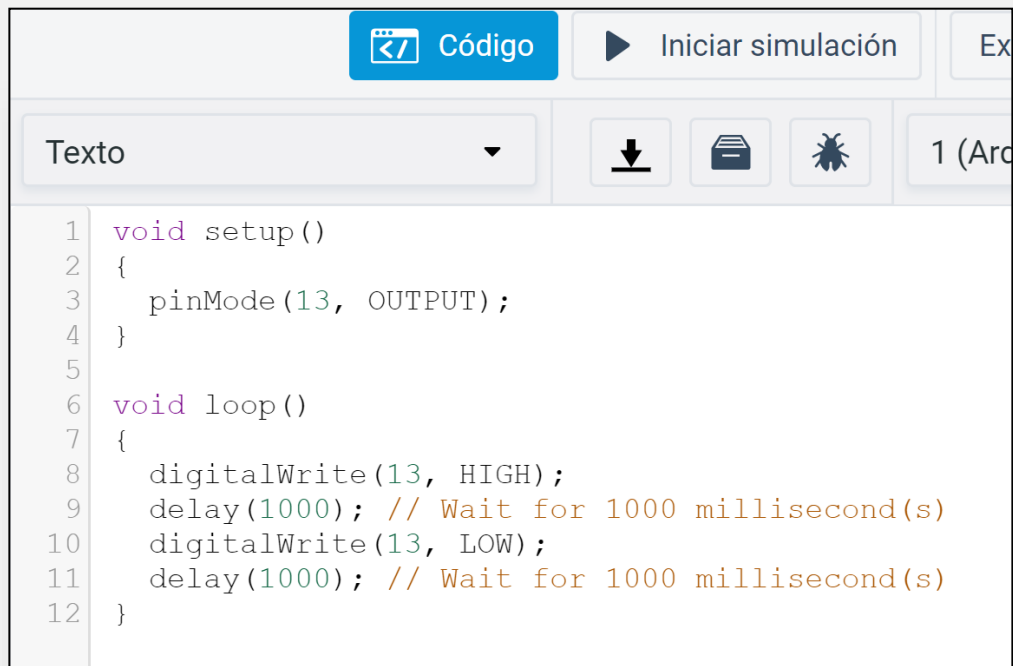


En el ejemplo...

En el `setup()` especificamos que usaremos el pin 13 como OUTPUT (un led).

En el `loop()` encendemos el pin 13, esperamos un segundo, apagamos el pin 13 y volvemos a esperar otro segundo... y la simulación volverá a iniciar (regresará a la primera línea dentro del `loop`).

Es por eso que durante la simulación vemos un led que se prende y apaga constantemente cada segundo.



The screenshot shows an IDE window with a toolbar at the top. The toolbar includes a 'Código' button with a code icon, an 'Iniciar simulación' button with a play icon, and a partially visible 'Ex' button. Below the toolbar is a dropdown menu labeled 'Texto' and a row of icons: a download arrow, a folder icon, a bug icon, and a page indicator '1 (Ar...'. The main area contains the following C++ code:

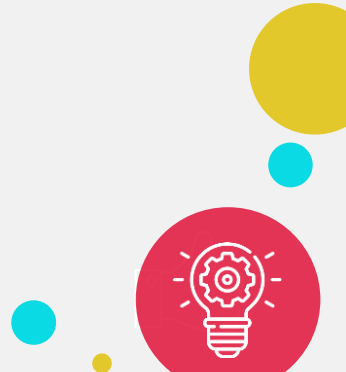
```
1 void setup()
2 {
3     pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8     digitalWrite(13, HIGH);
9     delay(1000); // Wait for 1000 millisecond(s)
10    digitalWrite(13, LOW);
11    delay(1000); // Wait for 1000 millisecond(s)
12 }
```


Reto

39

Configura el led para que muestre tu nombre.

MORSE CODE			
A	.-	H
B	-... .	I	..
C	-.-.- .	J	.- - - -
D	- . .	K	- . -
E	.	L	.- . .
F	.. - .	M	- -
G	- - .	N	- .
		O	- - -
		P	.- - - .
		Q	- - - .
		R	.- .
		S
		T	-
		U	.. -
		V	... -
		W	.- - -
		X	- . - .
		Y	- . - -
		Z	- - . .



Creado por:

Adriana Guadalupe Pastrana De la O

apastran@tec.mx

Estado de México

Ramiro Casas Gómez

rcasas@tec.mx

Eugenio Garza Lagüera

Alejandro Ehécatl Correa Cerón

alejandro.correacr@tec.mx

Valle Alto

Modificado por:

Luis Enrique S. García

luis.garcia@tec.mx

Hidalgo