

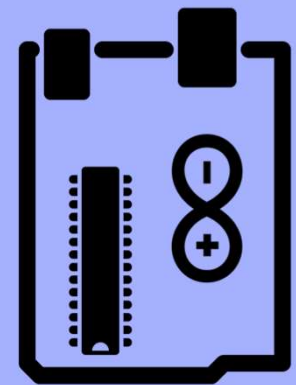
CURSO DE ARDUINO

MÓDULO 2



Por: Julián Andrés Castro
Estudiante ingeniería electrónica

CEIMTUN - RAS
UNIVERSIDAD NACIONAL DE COLOMBIA



Conócenos:

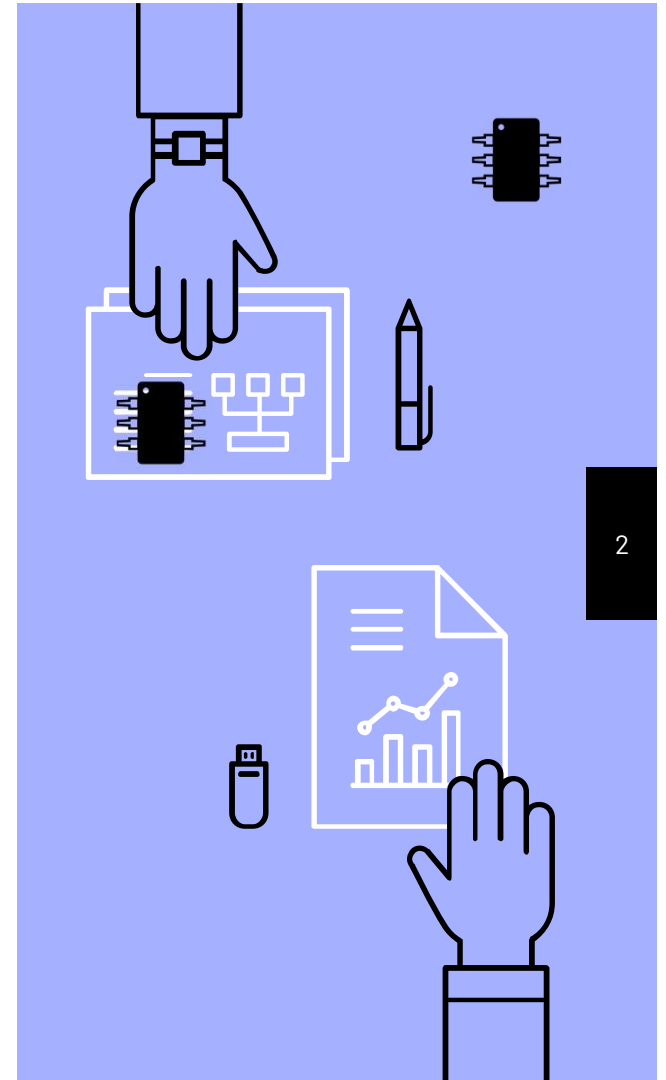
Web: [Ceimtun - RAS](http://Ceimtun-RAS)

Redes:

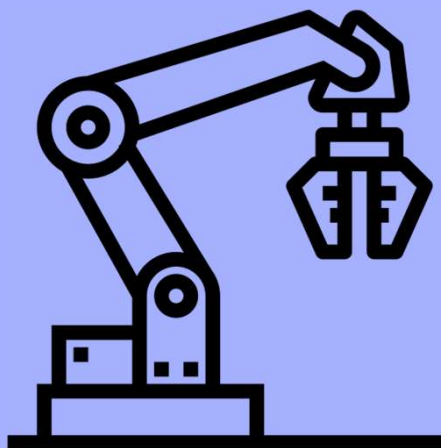
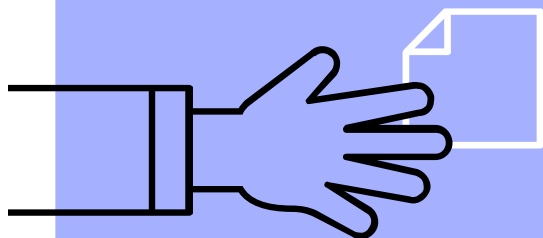


CONTENIDO: MÓDULO 2

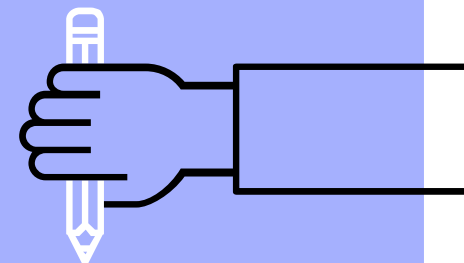
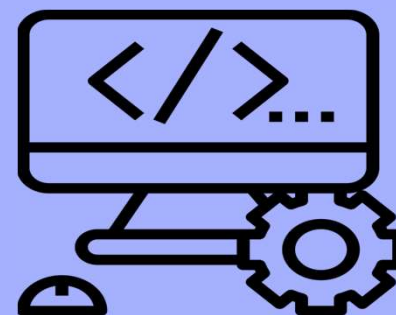
- ▶ Transductores / Actuadores
- ▶ Medición de temperatura con LM35
- ▶ Interruptor de luz con LDR
- ▶ Control de Motores con Módulo L298N
- ▶ Comunicación inalámbrica bluetooth



MÓDULO 2



JACP2



Diapositiva 3

JACP2

<https://www.flaticon.com/authors/becris>

JULIAN ANDRES CASTRO PARDO, 14/02/2019

TRANSDUCTORES Y ACTUADORES

Sistemas interactivos físicos valiéndose del uso de software y hardware para sensar y responder al mundo analógico.

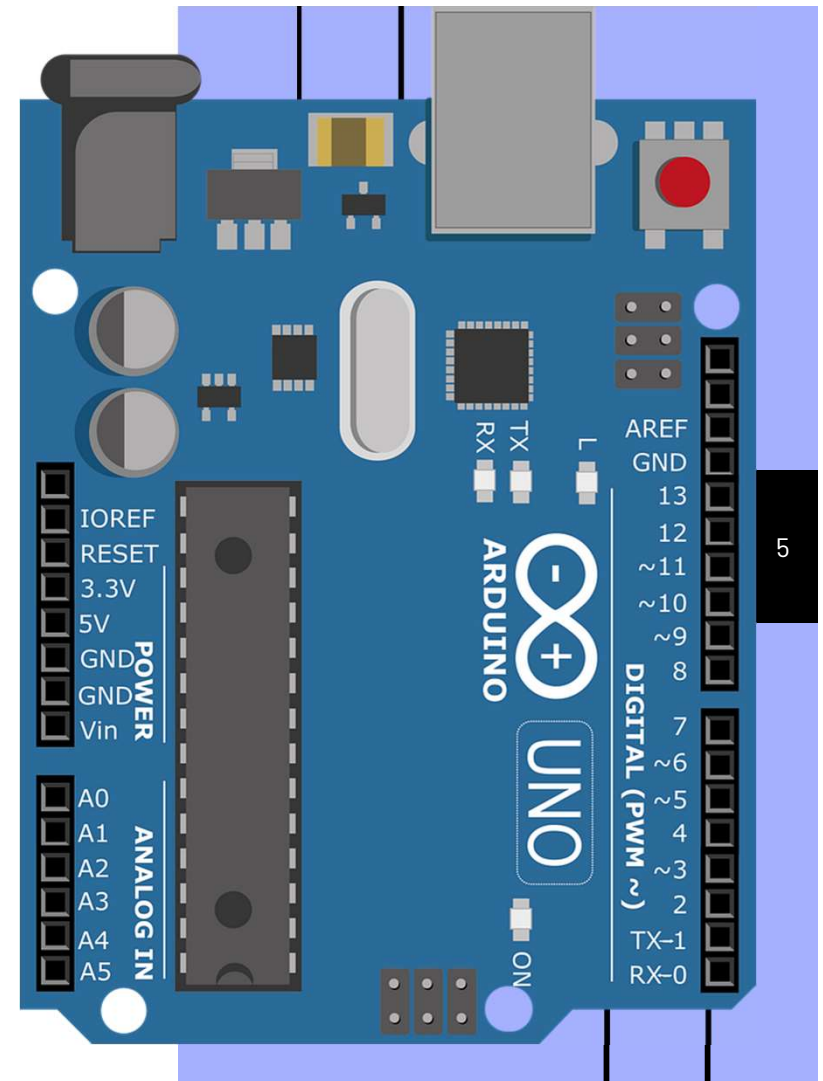
- Transforman magnitudes físicas en señales eléctricas y viceversa.
- **Transductores:** Sensores de temperatura, lumínicos, infrarrojo, etc.
- **Actuadores:** Motores, parlantes, bobinas, polímeros retroactivos, etc.



ARDUINO

- ▶ Plataforma de electrónica abierta para la creación de prototipos.
- ▶ Es de bajo costo, flexible y fácil de usar.
- ▶ Creada para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos.
- ▶ **JACP7**
- ▶ Lenguaje basado en Wiring / procesing

Historia Arduino: <https://arduinohistory.github.io/>



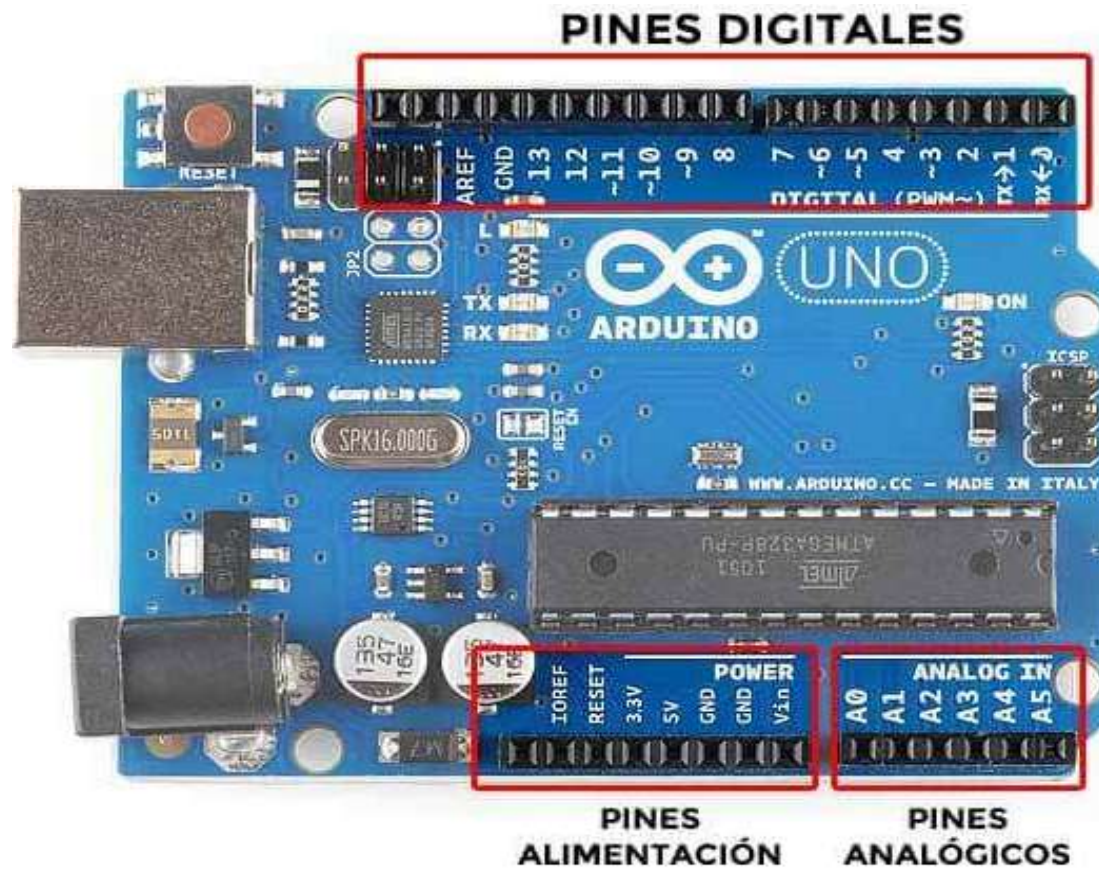
Diapositiva 5

JACP7

Basasdo en Wiring con entorno de desarrollo Processing

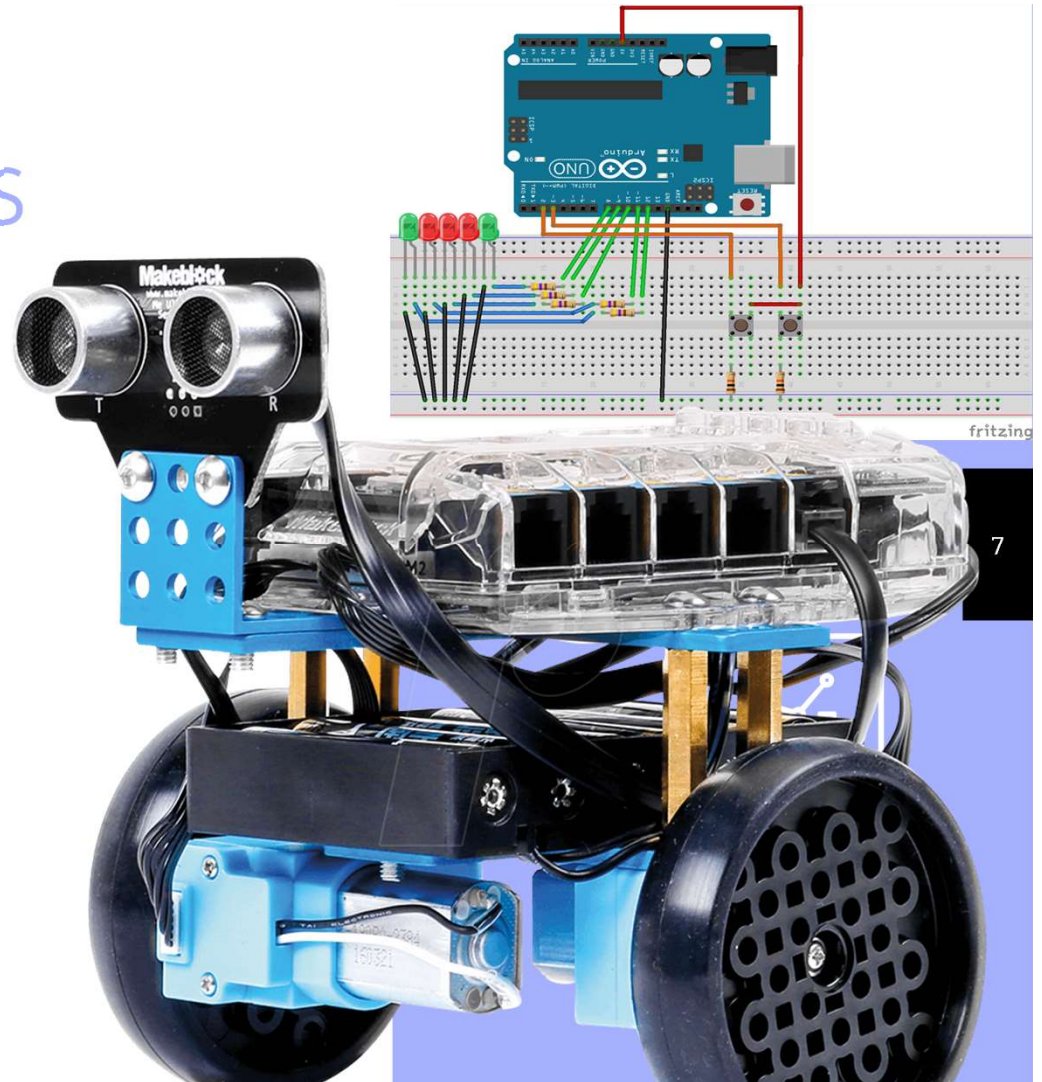
JULIAN ANDRES CASTRO PARDO, 15/02/2019

ARDUINO UNO



APLICACIONES

- ▶ Prototipos
- ▶ Juguetes
- ▶ Robótica simple
- ▶ Arte
- ▶ IoT - Internet of Things



ENTORNO DE DESARROLLO - IDE

HOME STORE SOFTWARE **EDU** RESOURCES COMMUNITY HELP

FAQ

CONTACT US

STORE SUPPORT

Download the Arduino IDE



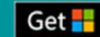
ARDUINO 1.8.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10



Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits

Linux 64 bits

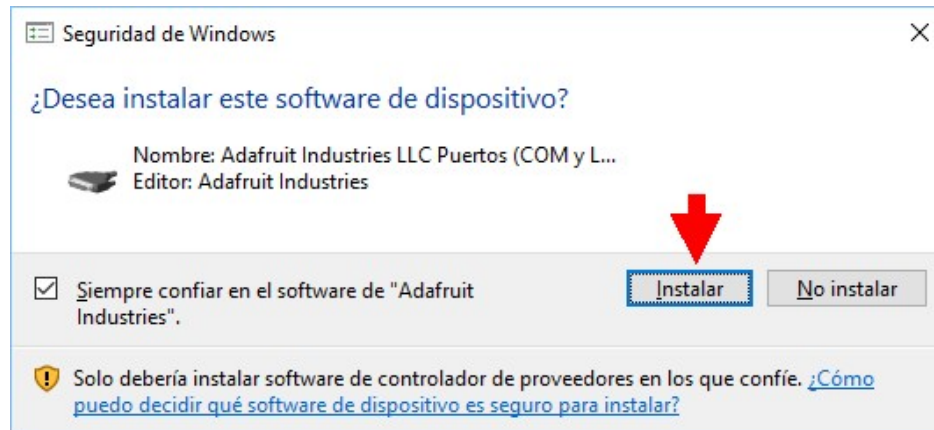
Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

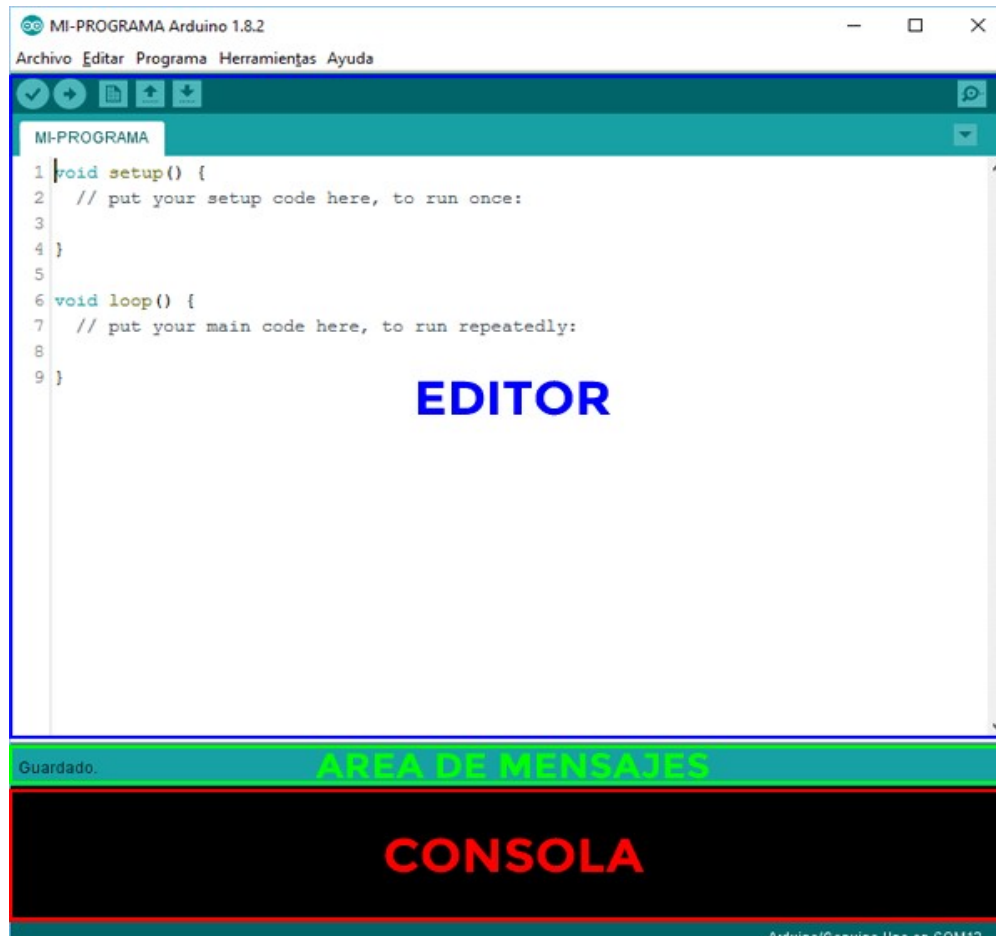
ENTORNO DE DESARROLLO - IDE



Instalar drivers

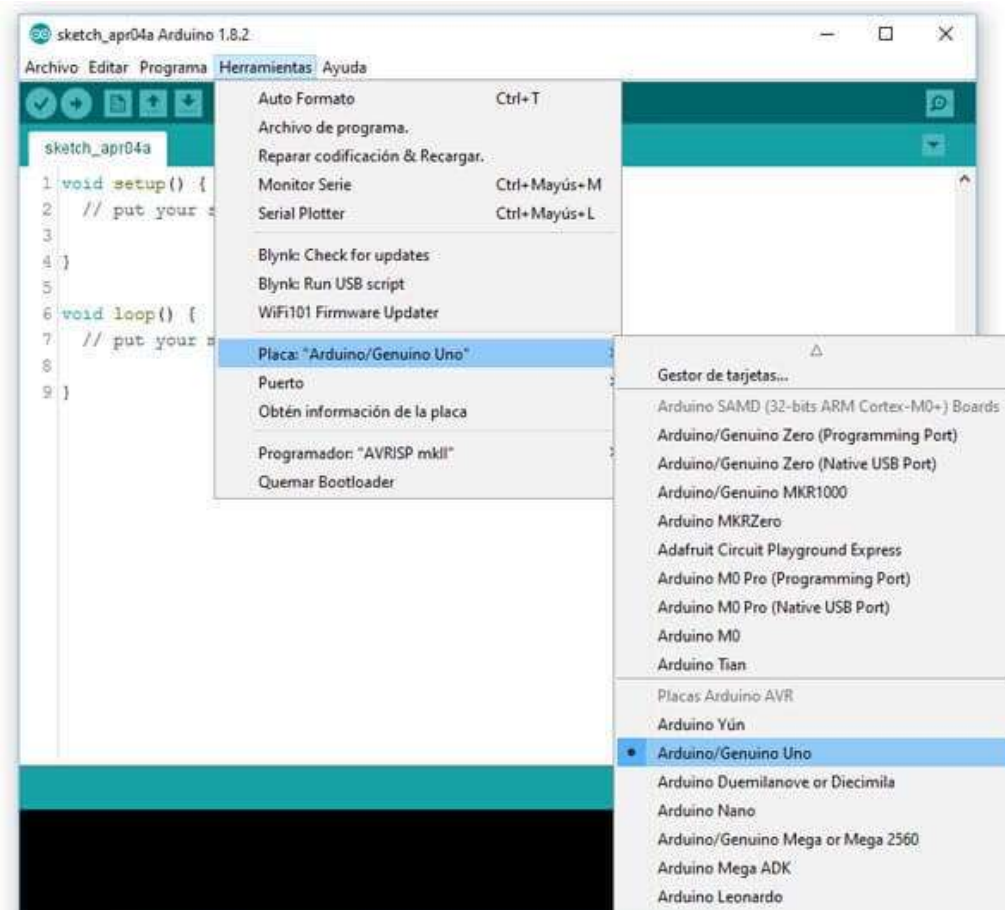


ENTORNO DE DESARROLLO - IDE



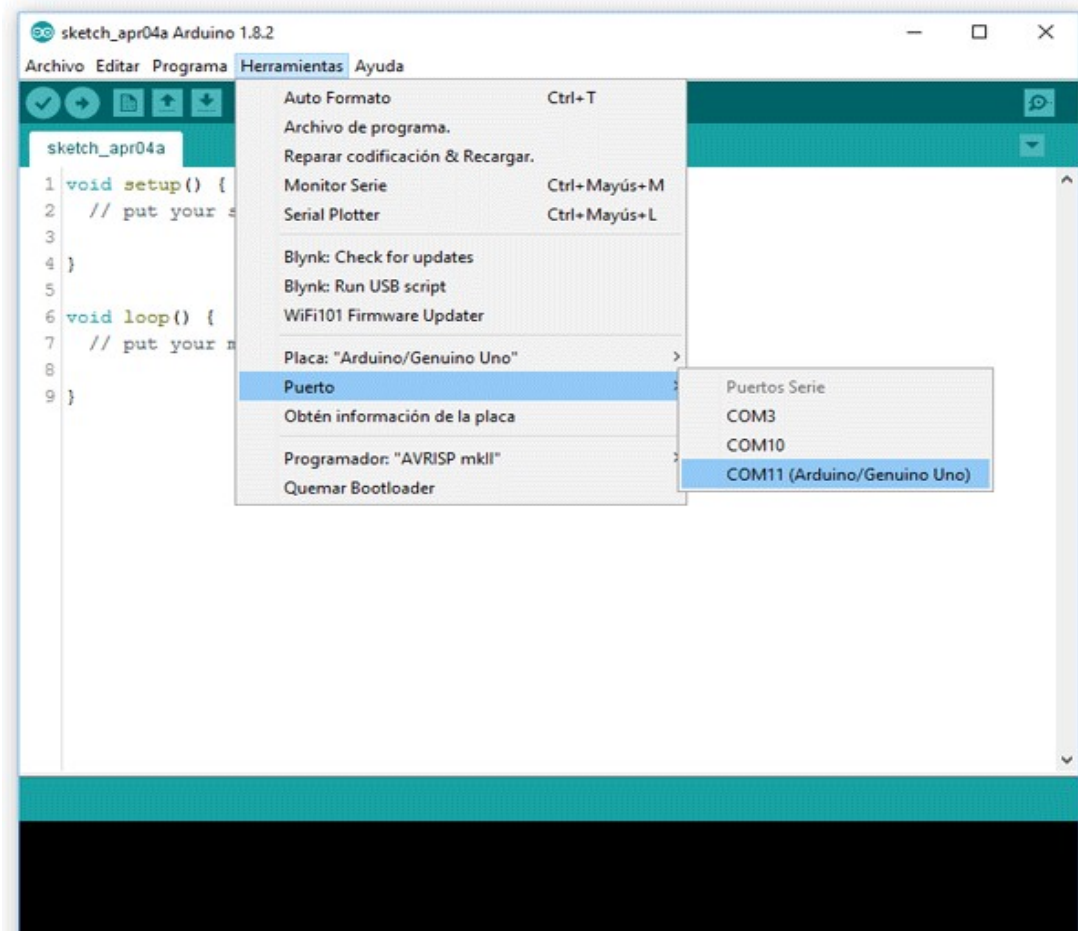
Partes fundamentales
del IDE de Arduino

ENTORNO DE DESARROLLO - IDE



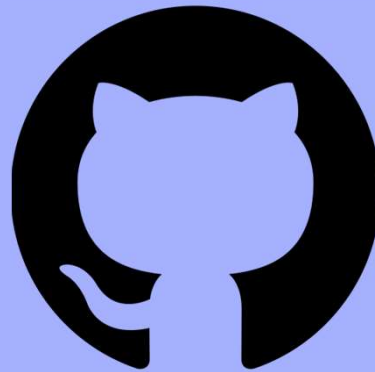
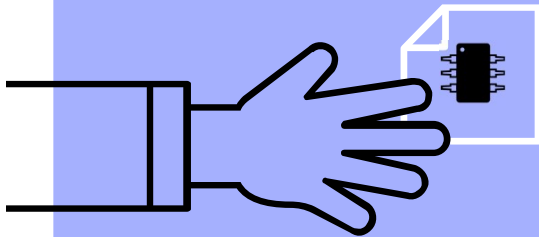
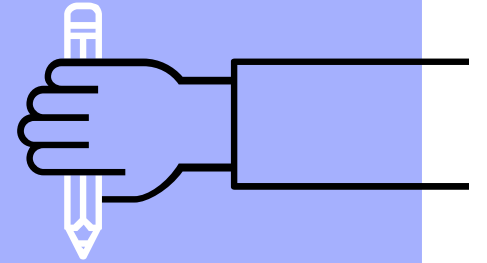
Seleccionar placa

ENTORNO DE DESARROLLO - IDE



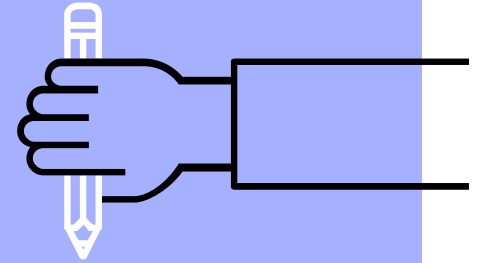
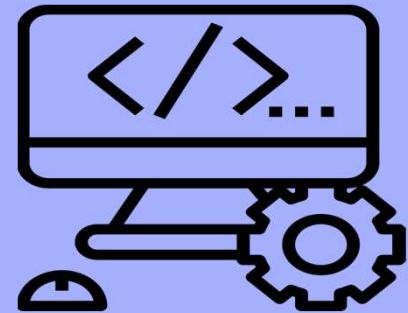
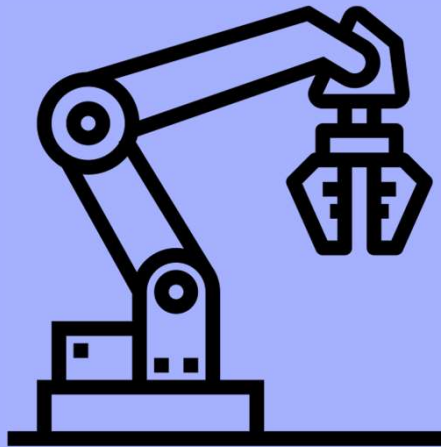
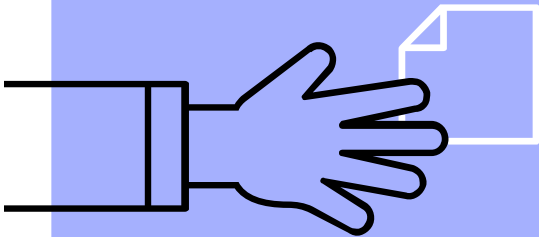
Seleccionar puerto

ACTIVIDADES

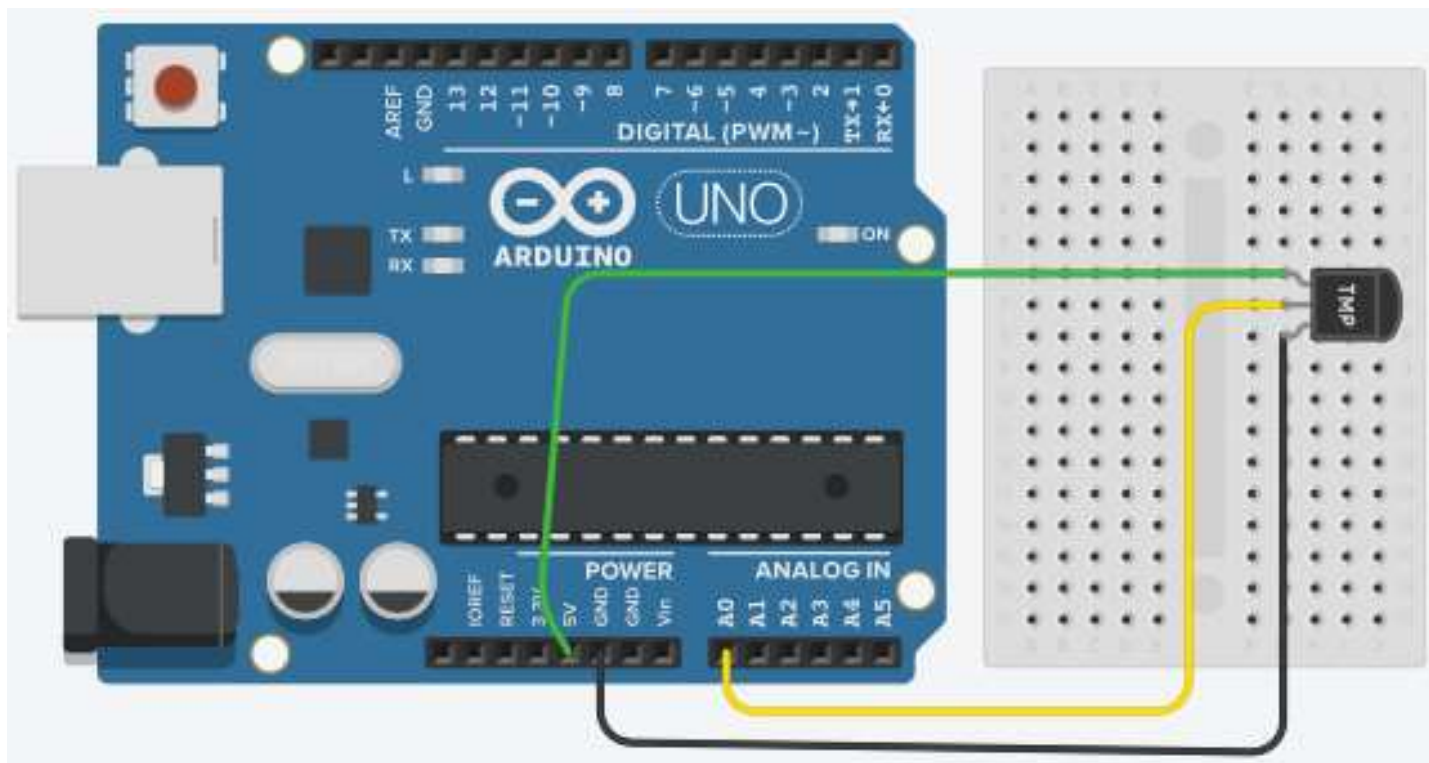


Descarga los Scripts
en [GitHub](#)!

SENSORES ANALÓGICOS

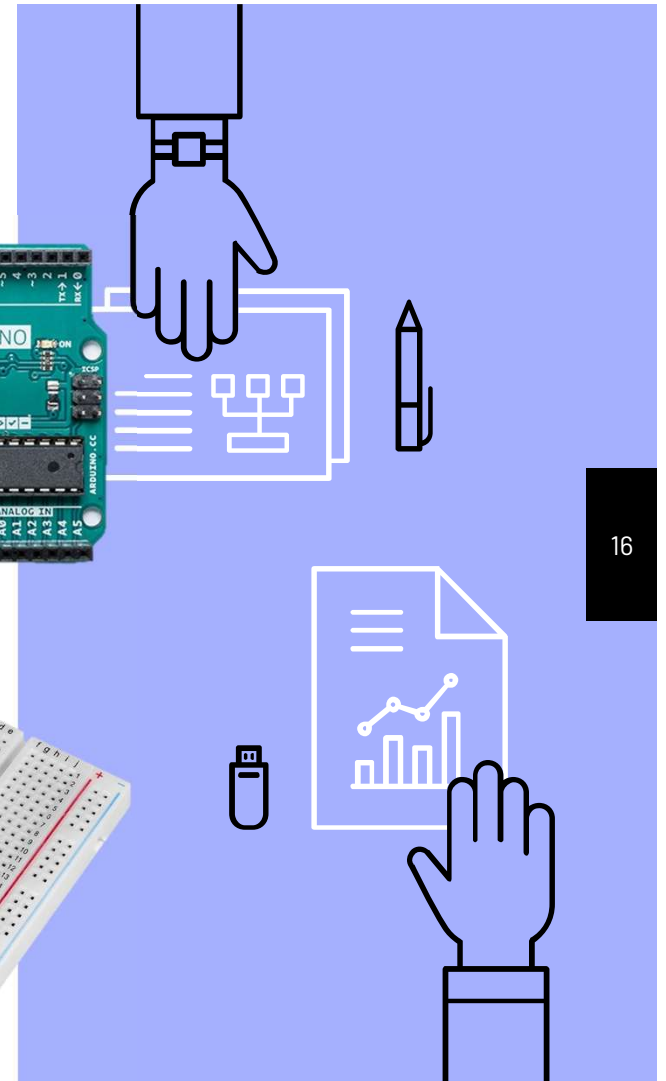
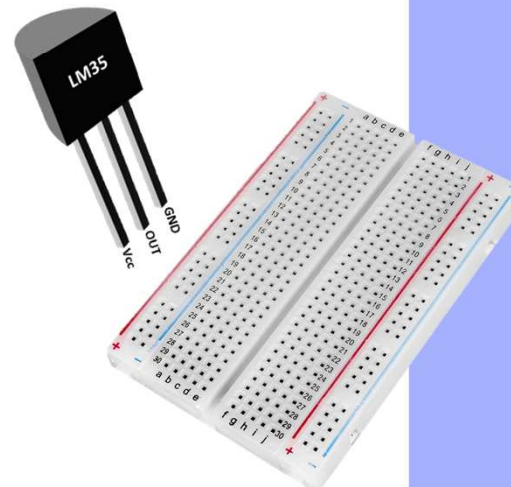


EJEMPLO 1: Medición de temperatura con LM35



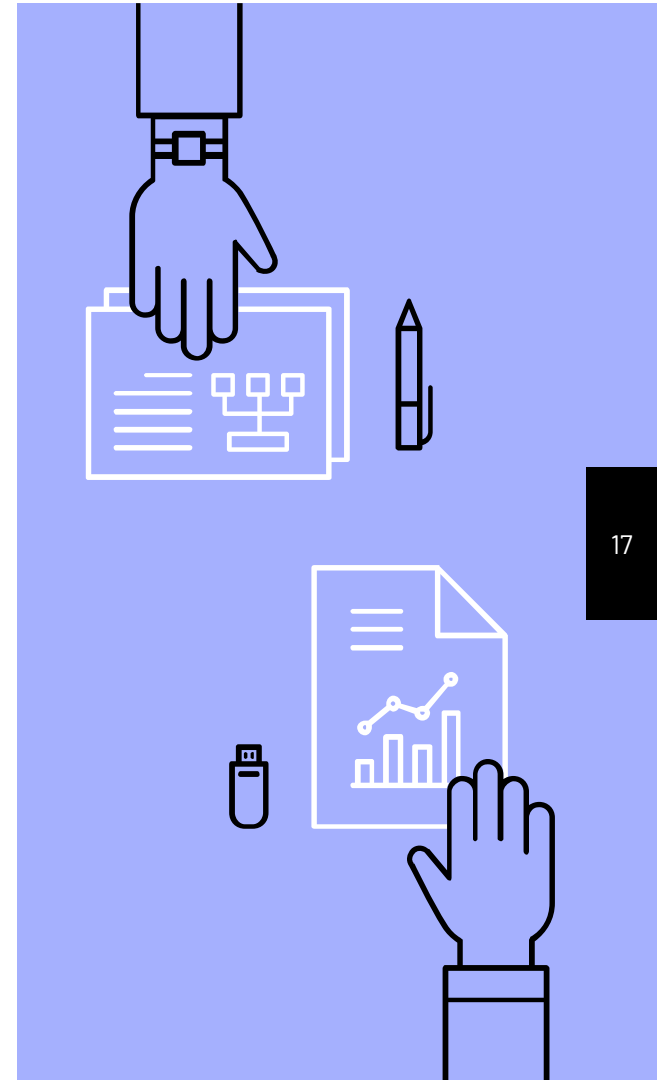
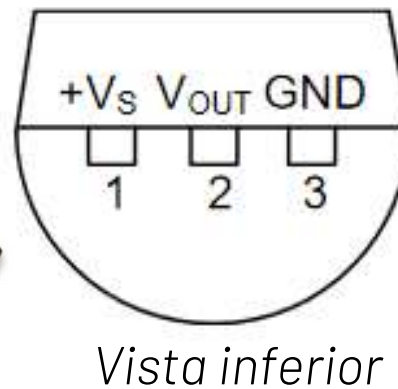
Materiales Ej. 1

- ▶ 1 Tarjeta Arduino con cable USB
- ▶ 1 Sensor de temperatura LM35
- ▶ 3 Jumpers macho-macho (cables)
- ▶ 1 Protoboard



Características del sensor LM35

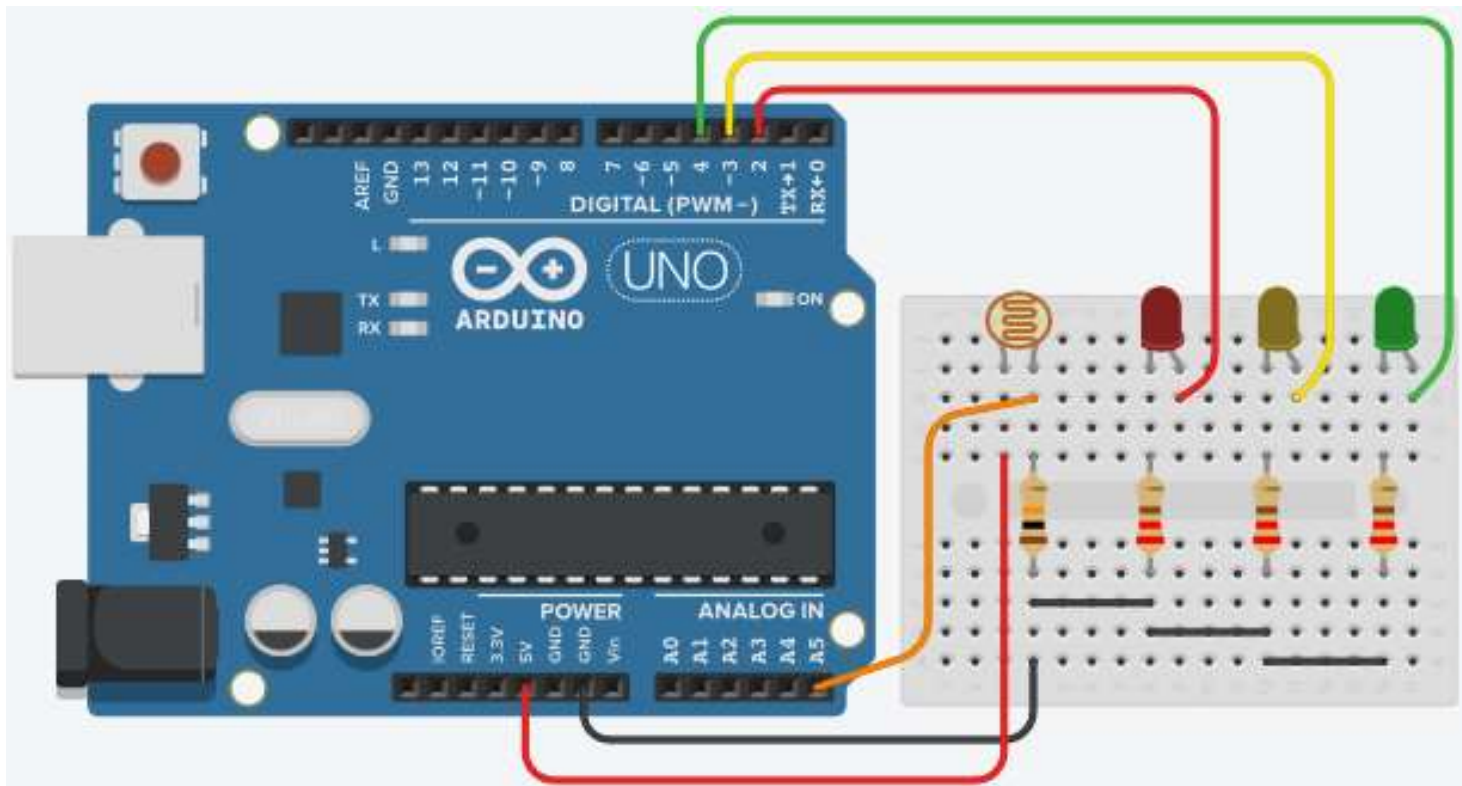
- ▶ Temperatura operación: $-55 \sim 150^{\circ}\text{C}$
- ▶ Alimentación: $4 \sim 30\text{V}$
- ▶ Salida: $-1 \sim 6\text{V}$
- ▶ Variación: $10\text{mV} / ^{\circ}\text{C}$
- ▶ Exactitud: $\pm 0.5^{\circ}\text{C}$
- ▶ Pinout:
 1. $+V_S$
 2. V_{OUT}
 3. GND



EJEMPLO 1: Medición de temperatura con LM35

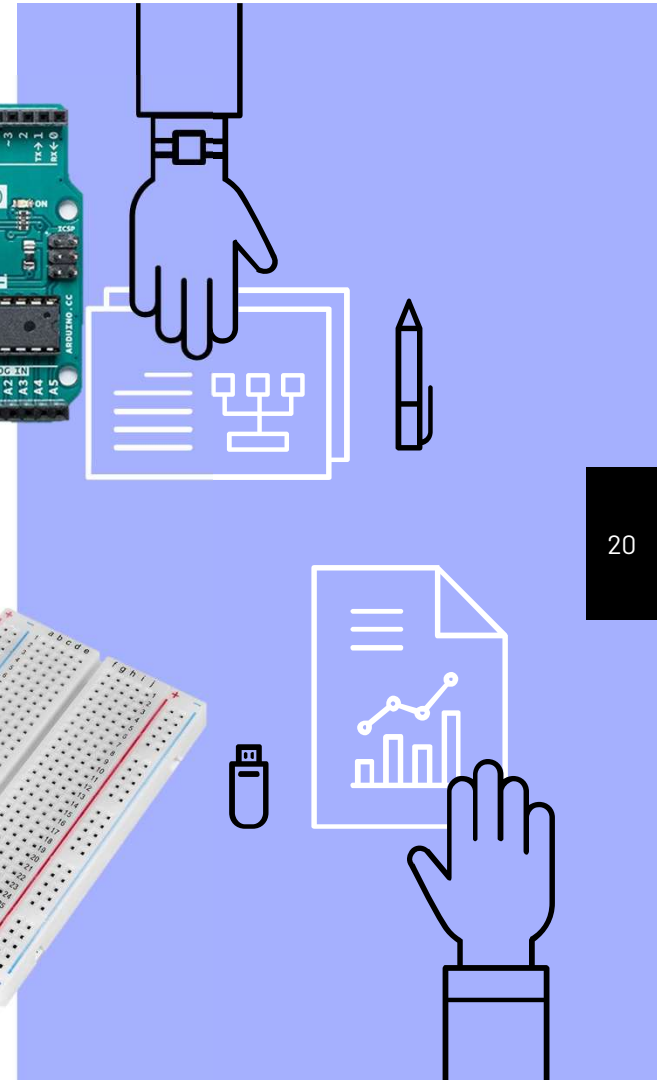
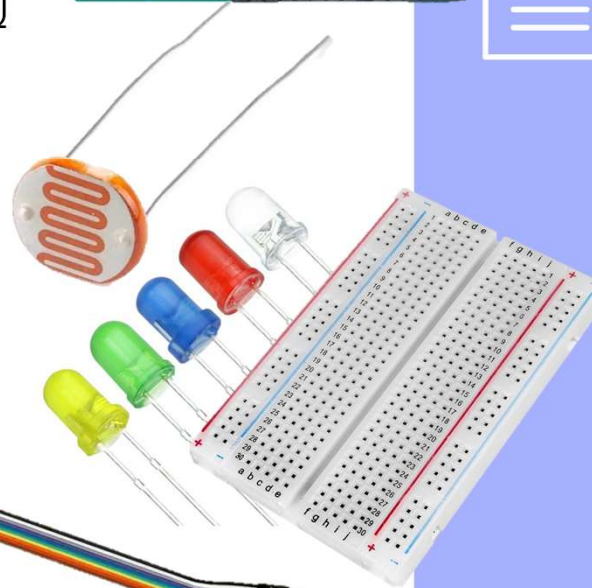
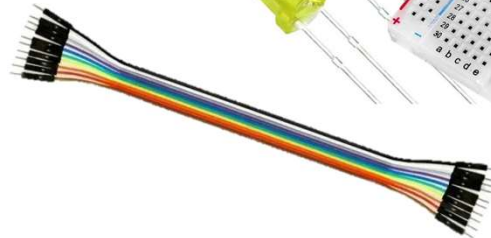
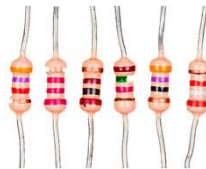
```
1 // Variables globales
2 float medida, temperaturaC; // Almacena el valor obtenido del sensor (0 a 1023)
3 int pinLM35 = A0;           // Pin de entrada del sensor (A0)
4 float t_ref = 100.0;        // float t_ref = 96.0; //°C
5
6 void setup() {
7     analogReference(INTERNAL); // Cambio de referencia entradas analógicas (1.1v)
8     Serial.begin(9600);        // Configura el puerto serial a 9600 bps
9 }
10
11 void loop() {
12     medida = analogRead(pinLM35); // analogRead lee el sensor (0-1023)
13     temperaturaC = (1.1 * medida * t_ref) / 1024.0; // Calcula la temperatura
14
15     Serial.print(1.1 * medida / 1024.0); // Envía el dato crudo
16     Serial.print(" mV ");
17     Serial.print(temperaturaC); // Envía el dato al puerto serial
18     Serial.println(" °C");
19
20     delay(1000); // Espera 1s para repetir el loop
21 }
```

EJEMPLO 2: Interruptor de luz con LDR



Materiales Ej. 2

- ▶ 1 Tarjeta Arduino
- ▶ 3 Leds 5mm (cualquier color)
- ▶ 3 Resistencias (220-330) Ω
- ▶ 1 Resistencias 10k Ω
- ▶ 1 Fotorresistor (LDR)
- ▶ 6 Jumpers macho-macho (cables)
- ▶ 1 Protoboard



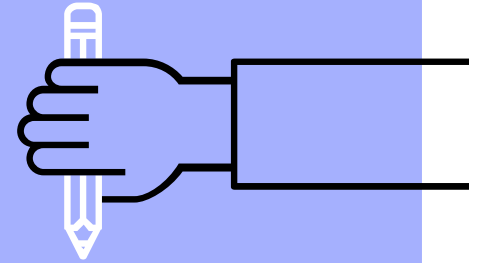
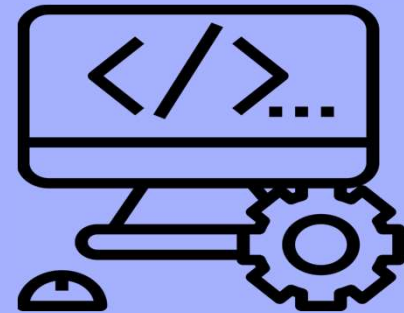
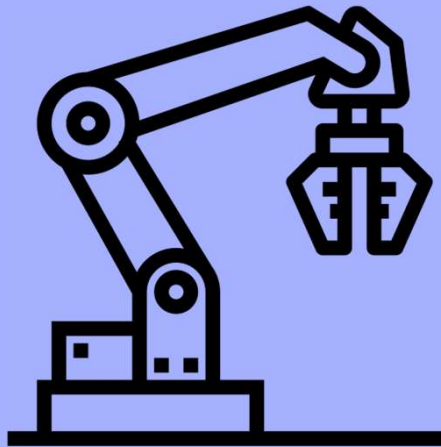
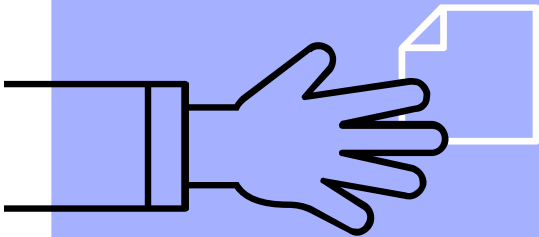
EJEMPLO 2: Interruptor de luz con LDR

```
1 // Variables globales
2 int pinLedG = 2;           // Pin conecta led Verde
3 int pinLedY = 3;           // Pin conecta led Amarillo
4 int pinLedR = 4;           // Pin conecta led Rojo
5 int pinLDR = A5;           // Pin analógico entrada del LDR
6 int valorLDR = 0;          // Almacena el valor del LDR
7 float voltajeLDR = 0.0;    // Almacena el valor del voltaje del LDR
8
9 void setup()
10 {
11     pinMode(pinLedR, OUTPUT); // Configura como salida el pin del led
12     pinMode(pinLedY, OUTPUT);
13     pinMode(pinLedG, OUTPUT);
14
15     Serial.begin(9600);       // Configura el puerto serial
16 }
```

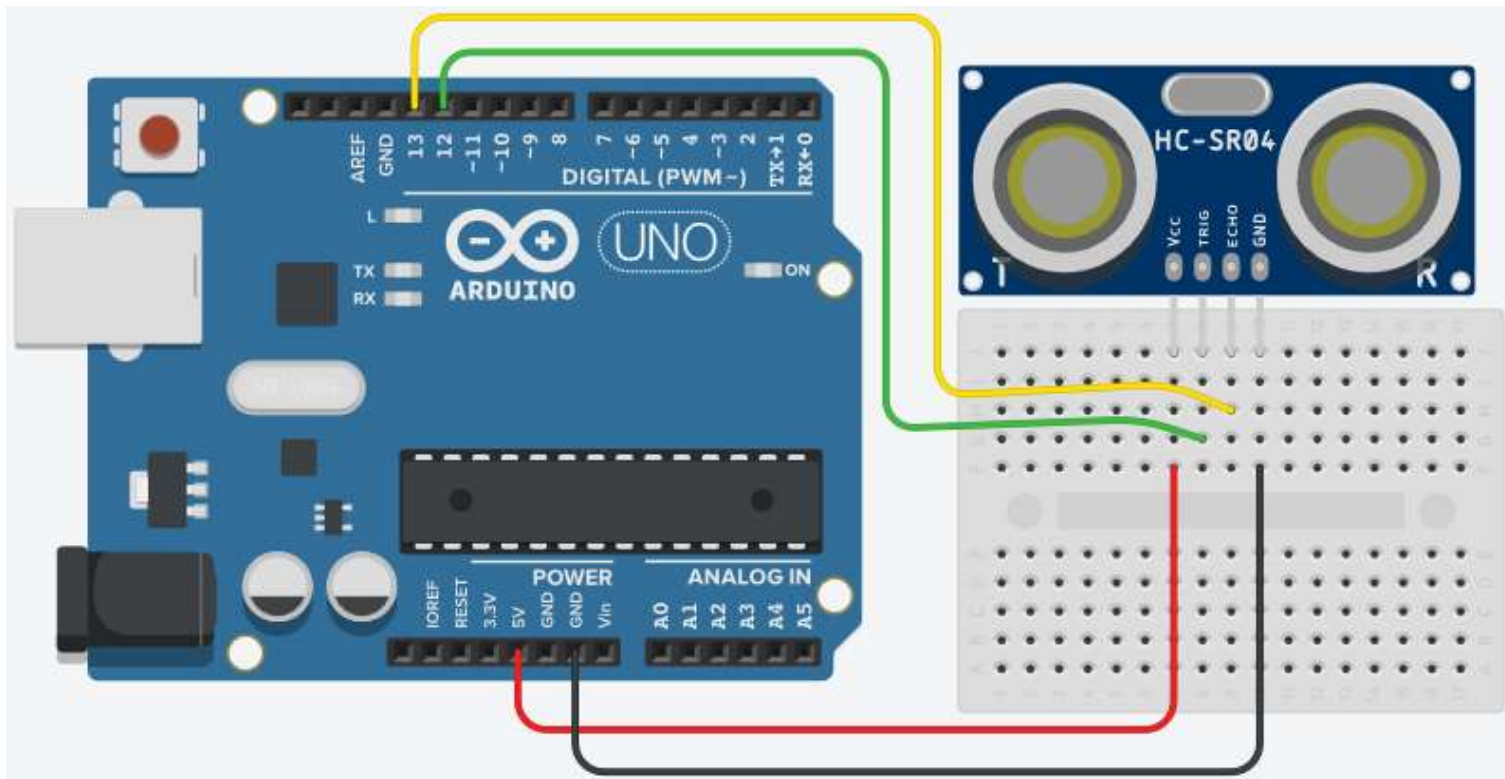
EJEMPLO 2: Interruptor de luz con LDR

```
17 void loop()
18 {
19     digitalWrite(pinLedR, LOW);    // Apaga el led R al iniciar el ciclo
20     digitalWrite(pinLedY, LOW);    // Apaga el led Y al iniciar el ciclo
21     digitalWrite(pinLedG, LOW);    // Apaga el led G al iniciar el ciclo
22     valorLDR = analogRead(pinLDR); // Lee el valor de voltaje del LDR
23     voltajeLDR = valorLDR * 5.0 / 1024.0; // Calcula el valor del voltaje
24
25     Serial.print(valorLDR); Serial.print(" ADC ");
26     Serial.print(voltajeLDR);      // Voltaje leído en el monitor serial
27     Serial.println(" V");
28
29     // Encender los leds de acuerdo al valor de ADC
30     if (valorLDR > 256.0) {
31         digitalWrite(pinLedR, HIGH);
32     }
33     if (valorLDR > 612.0) {
34         digitalWrite(pinLedY, HIGH);
35     }
36     if (valorLDR > 850.0) {
37         digitalWrite(pinLedG, HIGH);
38     }
39     delay(200);                    //Espera 200 mS antes de actualizar
40 }
```


SENSORES DIGITALES

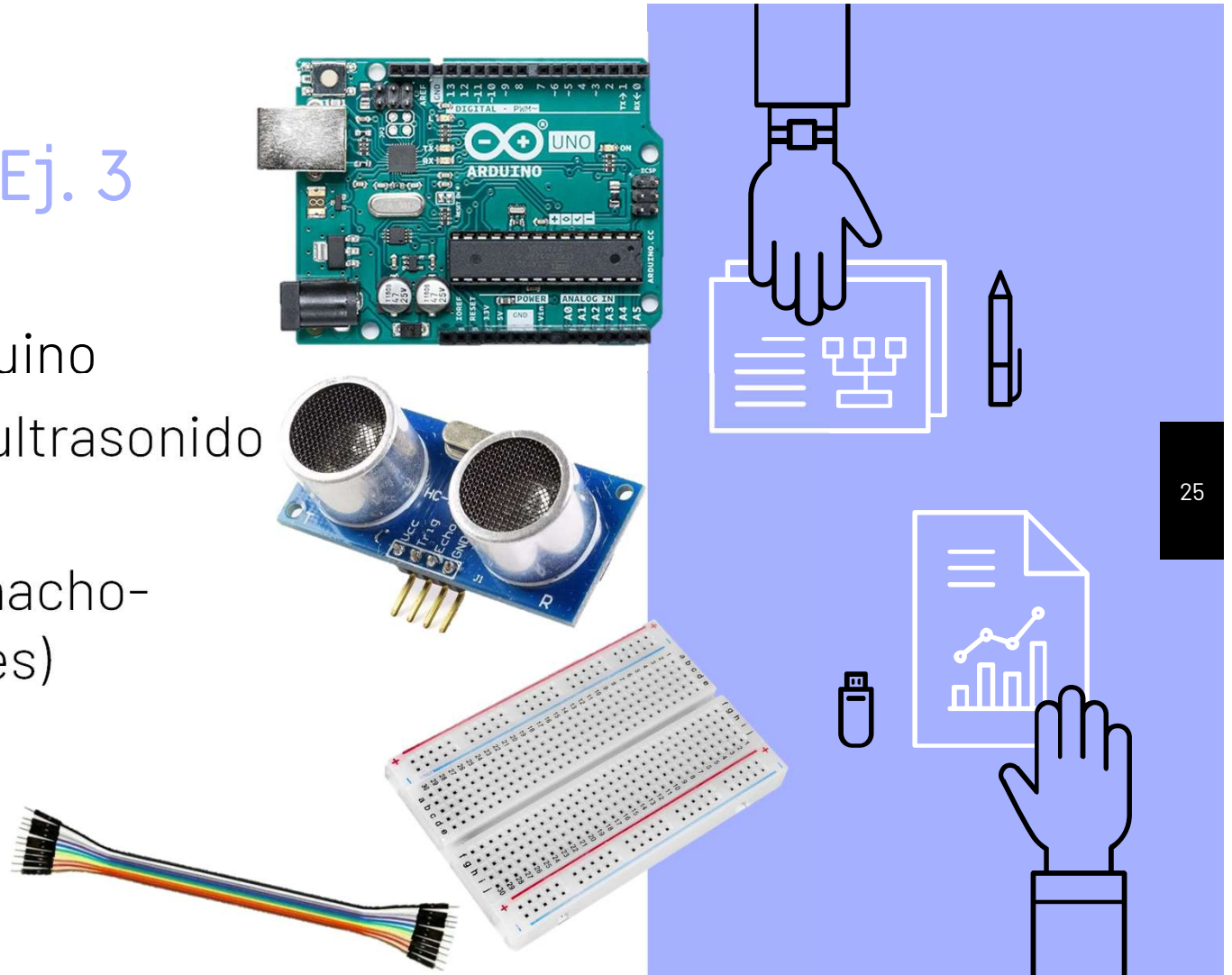


EJEMPLO 3: Medición de distancia con módulo de ultrasonido HC-SR04



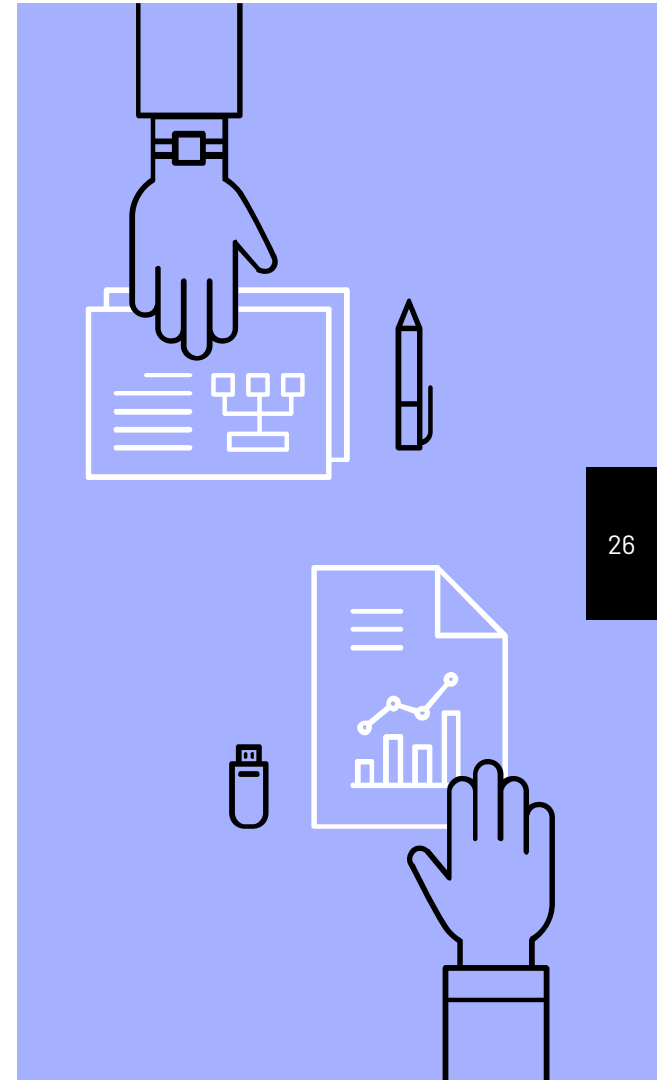
Materiales Ej. 3

- ▶ 1 Tarjeta Arduino
- ▶ 1 módulo de ultrasonido HC-SR04
- ▶ 4 Jumpers macho-macho (cables)
- ▶ 1 Protoboard



Características del sensor HC-SR04

- ▶ Rango de operación: 2cm ~ 4m
- ▶ Alimentación: 5V
- ▶ t de disparo (Pulso TTL): 10 μ s JACP13
- ▶ Resolución: ± 3 cm
- ▶ Pinout:
 1. +V
 2. Trig
 3. Echo
 4. GND



Diapositiva 26

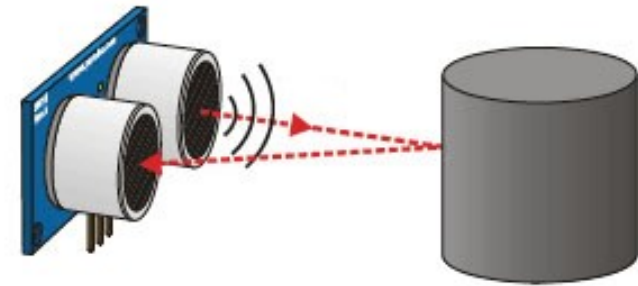
JACP13

TTL: Transistor - Transistor - Logic

JULIAN ANDRES CASTRO PARDO, 26/06/2020

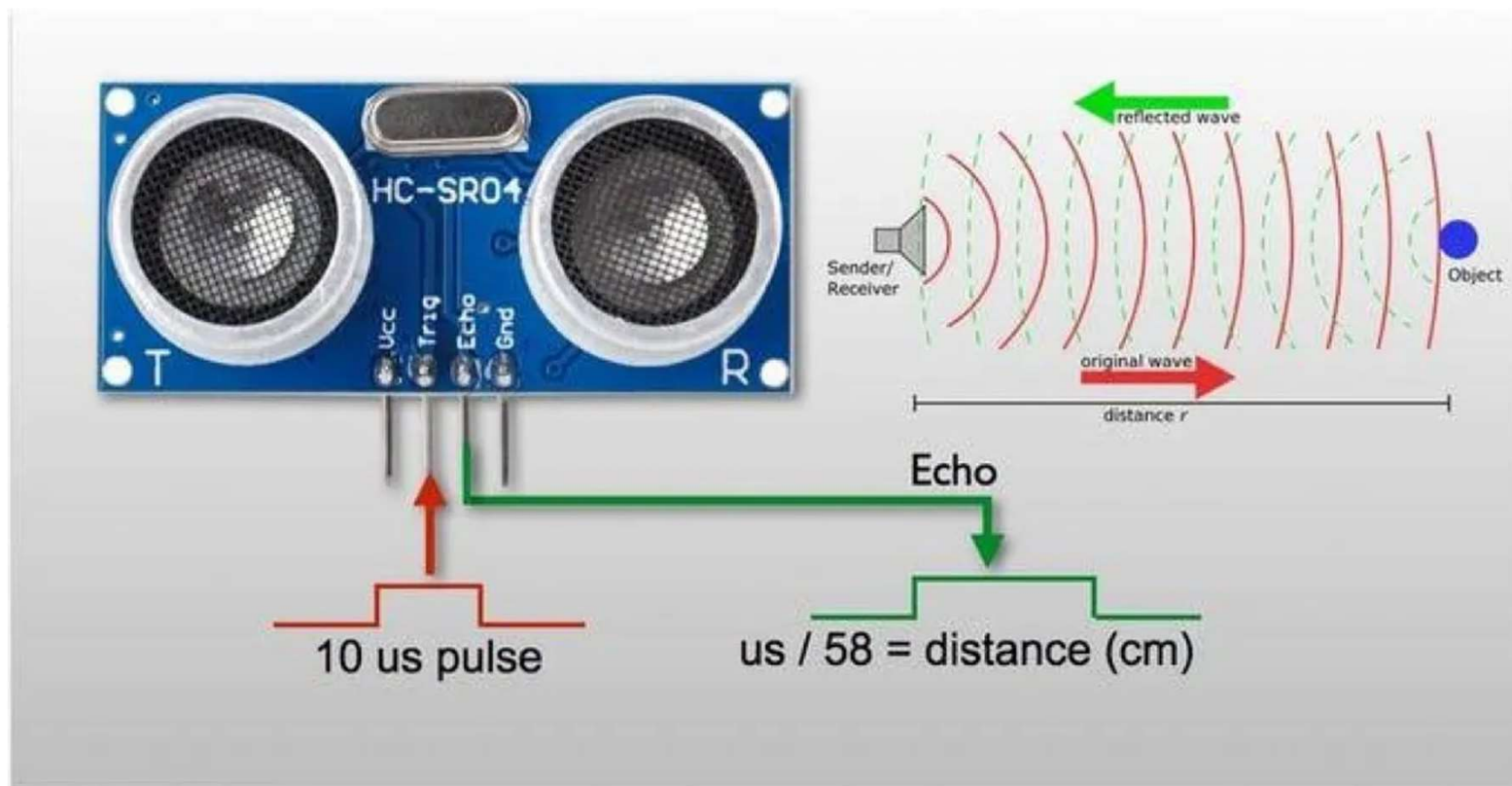
EJEMPLO 3: Medición de distancia con módulo de ultrasonido HC-SR04

- ▶ Se mide el tiempo que pasa entre el envío y la recepción del pulso.
- ▶ Velocidad del sonido: $V_s = 343 \text{ m/s}$
- ▶ $343 \text{ m/s} = 100 \text{ cm/m} * \frac{1}{1000000 \mu s} = \frac{1}{29.2 \mu s/cm}$
- ▶ $Distancia(cm) = \frac{Tiempo(\mu s)}{2 * 29.2 \mu}$



$$\begin{aligned} \text{Tiempo} &= 2 * (\text{Distancia} / \text{Velocidad}) \\ \text{Distancia} &= \text{Tiempo} \cdot \text{Velocidad} / 2 \end{aligned}$$

Características del sensor HC-SR04



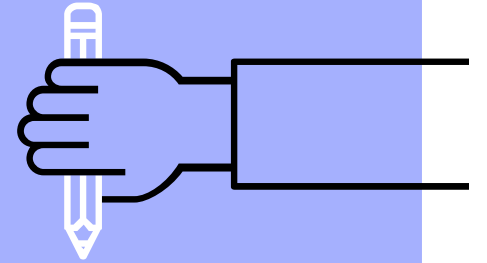
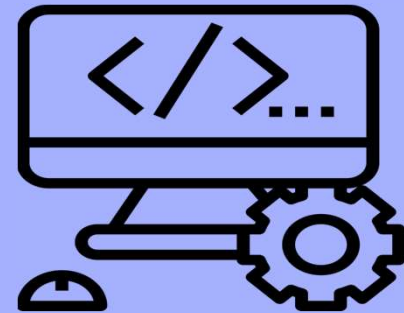
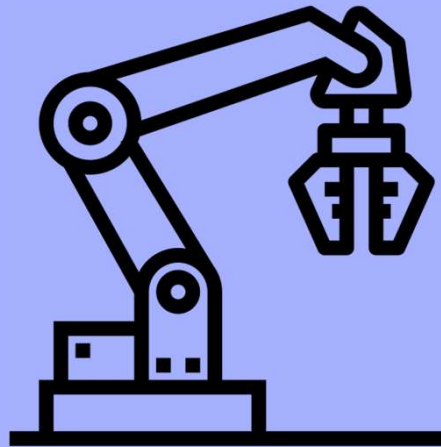
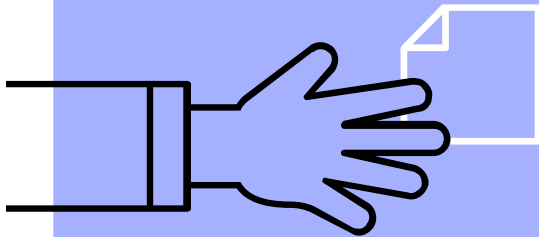
EJEMPLO 3: Medición de distancia con módulo de ultrasonido HC-SR04

```
1 // Variables globales
2 const int pinTrig = 12;           // Pin de disparo del ultrasonido
3 const int pinEco = 13;
4 const long tiempoEspera = 30000; // (30ms = 5m)~(3ms = 50cm)
5 int distancia;                   // Pin de recepción del eco
6
7 void setup() {
8     pinMode(pinTrig, OUTPUT);
9     pinMode(pinEco, INPUT);
10    Serial.begin(9600);           // Inicializa comunicación serial a 9600 bps
11 }
```

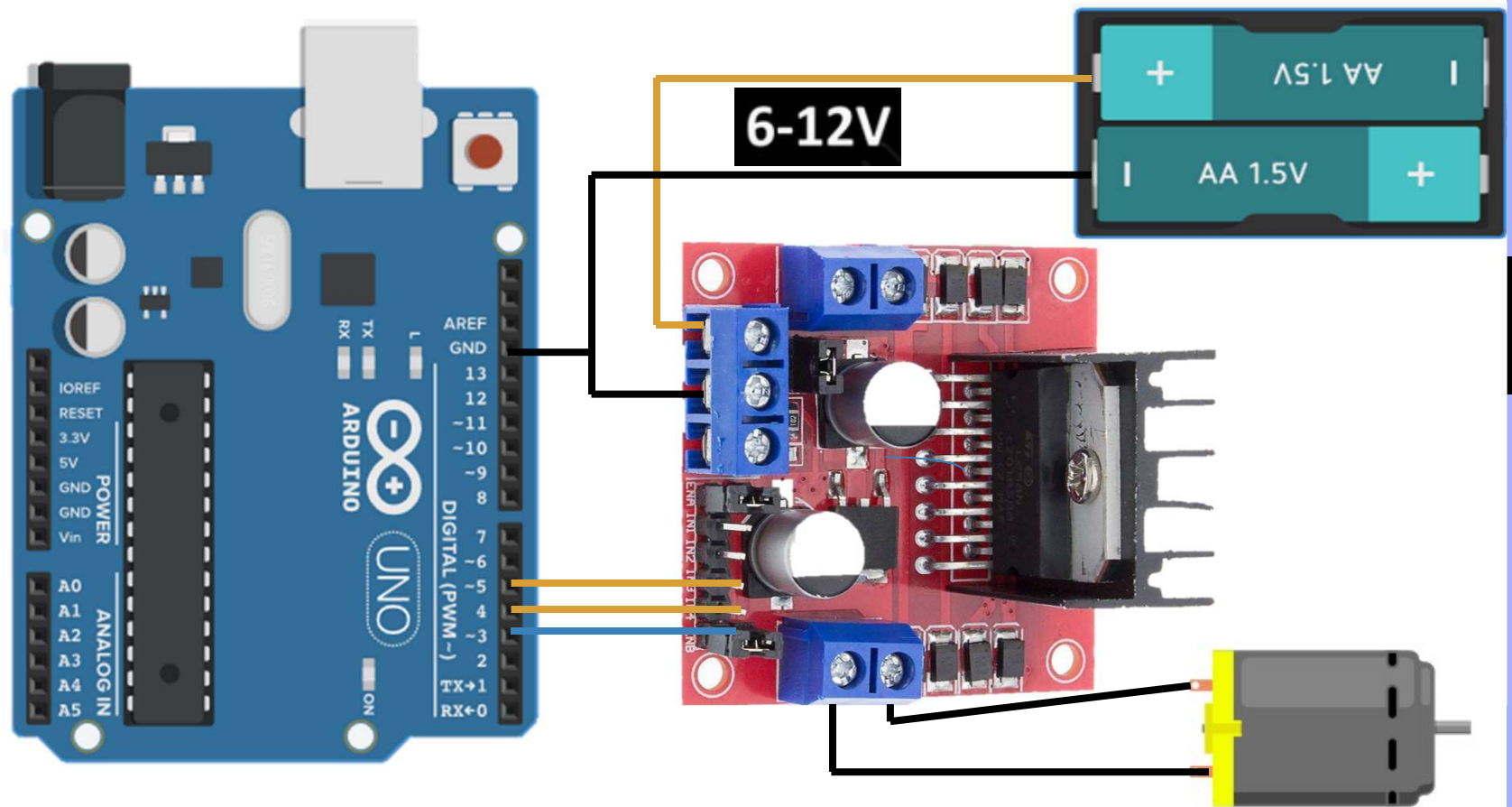

EJEMPLO 3: Medición de distancia con HC-SR04

```
12 void loop() {  
13     distancia = medirDistancia();           // Mide la distancia en cm  
14     Serial.print("Distancia: ");  
15     Serial.print(distancia);               // Visualización en monitor serie  
16     Serial.println(" cm");  
17     delay(500);  
18 }  
19  
20 long medirDistancia() {  
21     long tiempoPulso, distancia_cm;  
22     digitalWrite(pinTrig, LOW);           // Limpiar el disparador  
23     delayMicroseconds(4);  
24     digitalWrite(pinTrig, HIGH);  
25     delayMicroseconds(10);                // Tiempo del disparo 10 mS  
26     digitalWrite(pinTrig, LOW);  
27     tiempoPulso = pulseIn(pinEco, HIGH);  // Lee la duración de un pulso  
28 if ( tiempoPulso == 0 ) {  
29     tiempoPulso = tiempoEspera;  
30 }  
31     distancia_cm = tiempoPulso / 29 / 2;   // Calcula la distancia  
32     return distancia_cm;  
33 }
```

CONTROL DE MOTORES



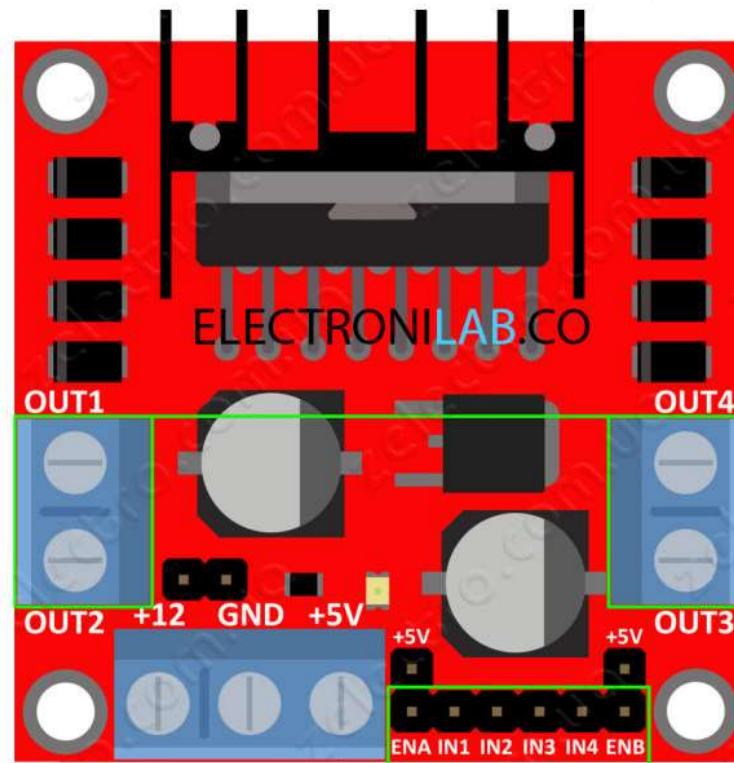
EJEMPLO 4: Control de Motores con Módulo L298N



EJEMPLO 4: Control de Motores con Módulo L298N

► Control PWM.

► Puente H



Salidas inabilitadas



Salidas habilitadas

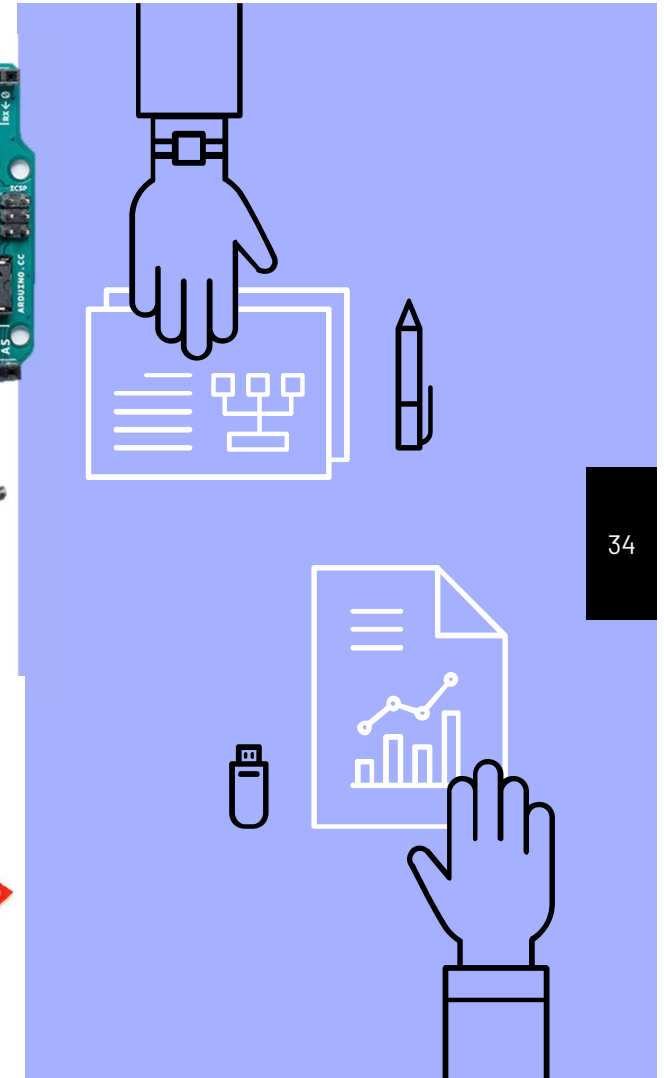
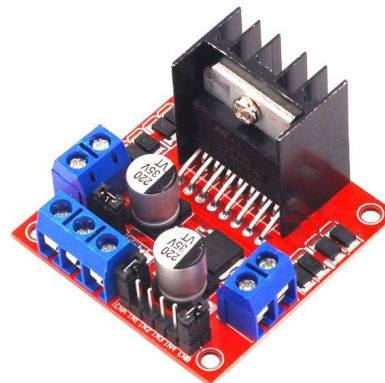


Salidas para motores

Terminales de control

Materiales Ej. 4

- ▶ 1 Tarjeta Arduino
- ▶ 1 motor DC 3-6V
- ▶ 1 Módulo Driver motor L298N
- ▶ 1 Potenciómetro (1-10k) Ω
- ▶ 9 Jumpers macho-macho
- ▶ 3 Jumpers macho - hembra



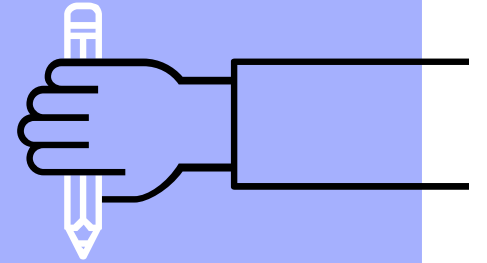
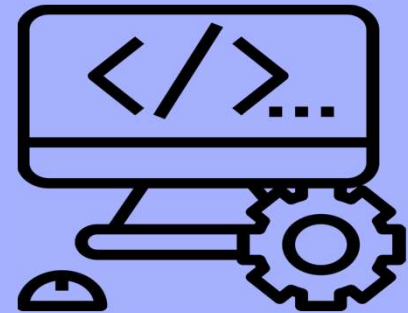
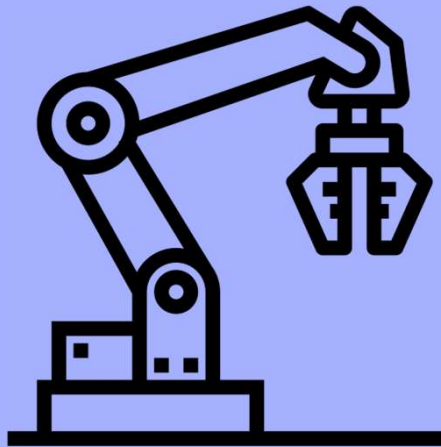
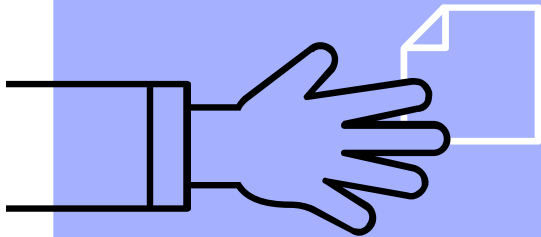
EJEMPLO 4: Control de Motores con Módulo L298N

```
1 int IN3 = 5;    // Input3 conectada al pin 5
2 int IN4 = 4;    // Input4 conectada al pin 4
3 int ENB = 3;    // ENB conectada al pin 3 de Arduino
4 void setup() {
5     Serial.begin(9600); // Inia comunicación serial
6     pinMode (ENB, OUTPUT);
7     pinMode (IN3, OUTPUT);
8     pinMode (IN4, OUTPUT);
9
10 }
```

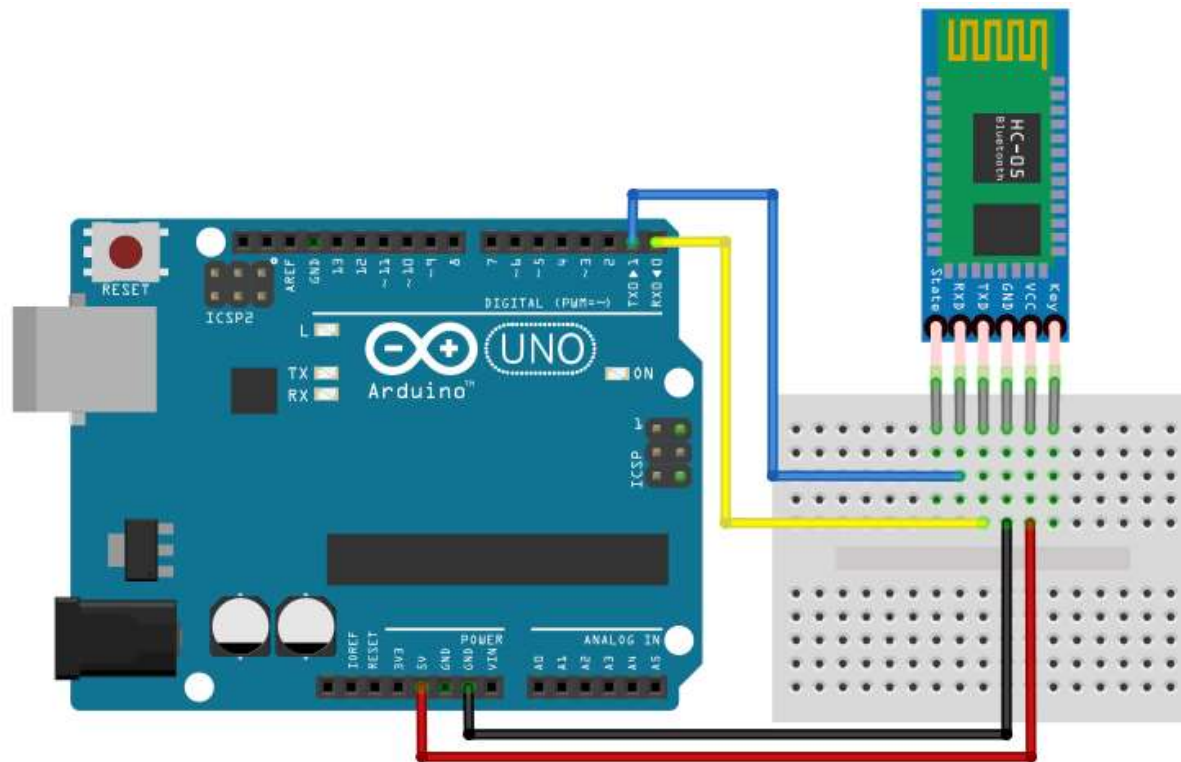
EJEMPLO 4: Control de Motores con Módulo L298N

```
11 void loop() {  
12     //Preparamos la salida para que el motor gire en un sentido  
13     digitalWrite (IN3, HIGH);  
14     digitalWrite (IN4, LOW);  
15     // Aplicamos PWM al pin ENB, haciendo girar el motor, cada 2 seg aumenta la velocidad  
16     analogWrite(ENB, 55);  
17     Serial.print("Velocidad PWM: ");  
18     Serial.println(55);  
19     delay(2000);  
20     analogWrite(ENB, 105);  
21     Serial.print("Velocidad PWM: ");  
22     Serial.println(105);  
23     delay(2000);  
24     analogWrite(ENB, 255);  
25     Serial.print("Velocidad PWM: ");  
26     Serial.println(255);  
27     delay(2000);  
28     // Apagamos el motor y esperamos 5 seg  
29     analogWrite(ENB, 0);  
30     delay(5000);  
31 }
```


COMUNICACIÓN INALÁMBRICA

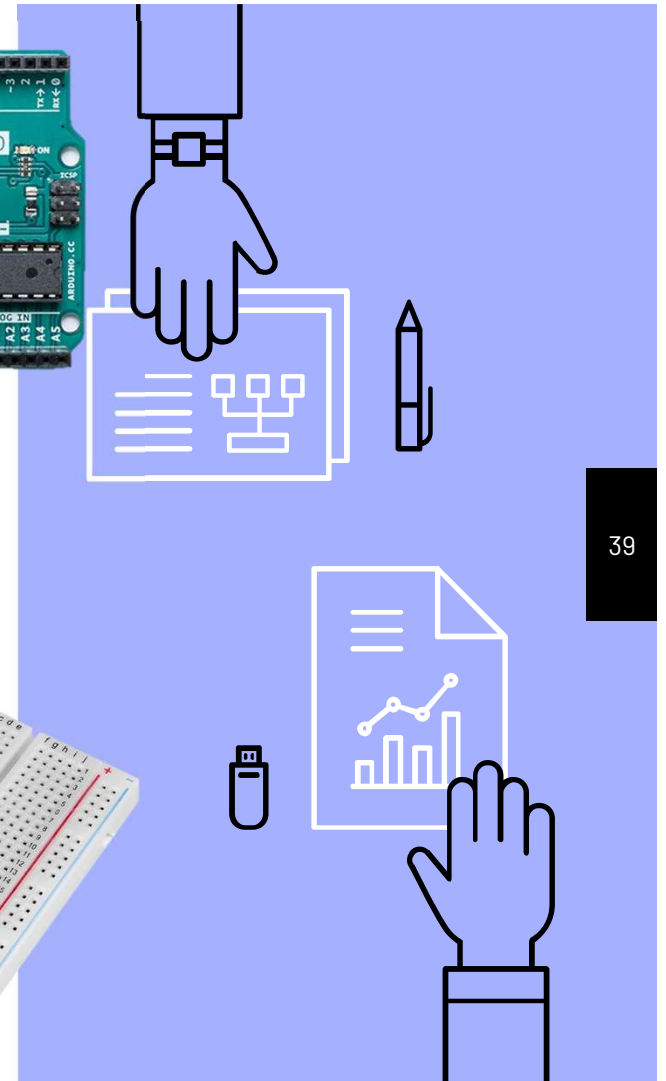
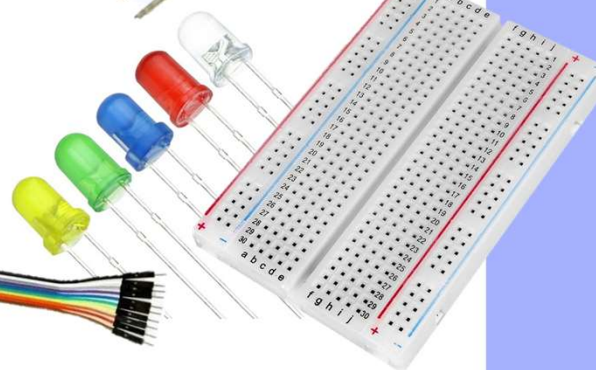
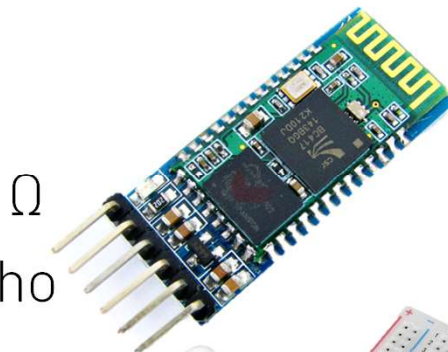
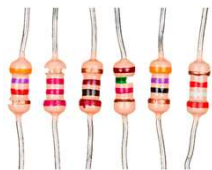


EJEMPLO 5: Comunicación inalámbrica bluetooth con módulo HC-05 o HC-06.



Materiales Ej. 5

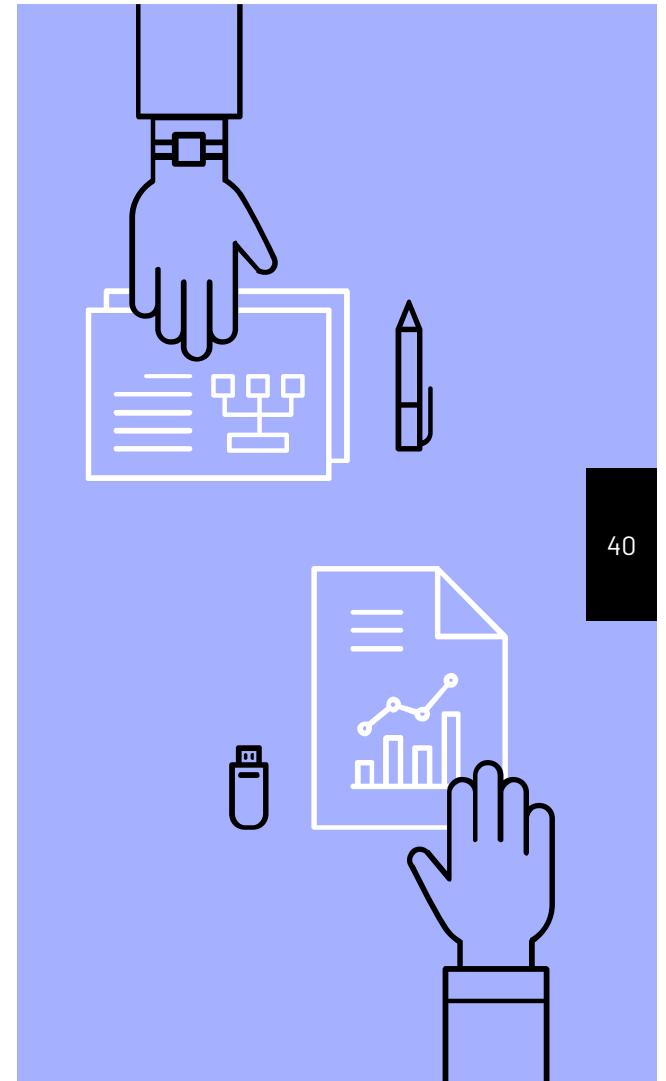
- ▶ 1 Tarjeta Arduino
- ▶ 1 Módulo bluetooth HC-05 o HC-06.
- ▶ 1 Led (cualquier color)
- ▶ 1 Resistencia (220-330) Ω
- ▶ 7 Jumpers macho-macho (cables)
- ▶ 1 Protoboard



Pasos para establecer comunicación

- ▶ Paso 1: alimente el Bluetooth
- ▶ Conecte el dispositivo en su teléfono (Clave: 1234)
- ▶ Abra la aplicación *Serial Bluetooth Terminal*
- ▶ Menú >Pestaña *Devices*: Seleccionar HC-05

***¡Advertencia!** No cargar programas al Arduino mientras el bluetooth está conectado*



EJEMPLO 5: Comunicación inalámbrica bluetooth

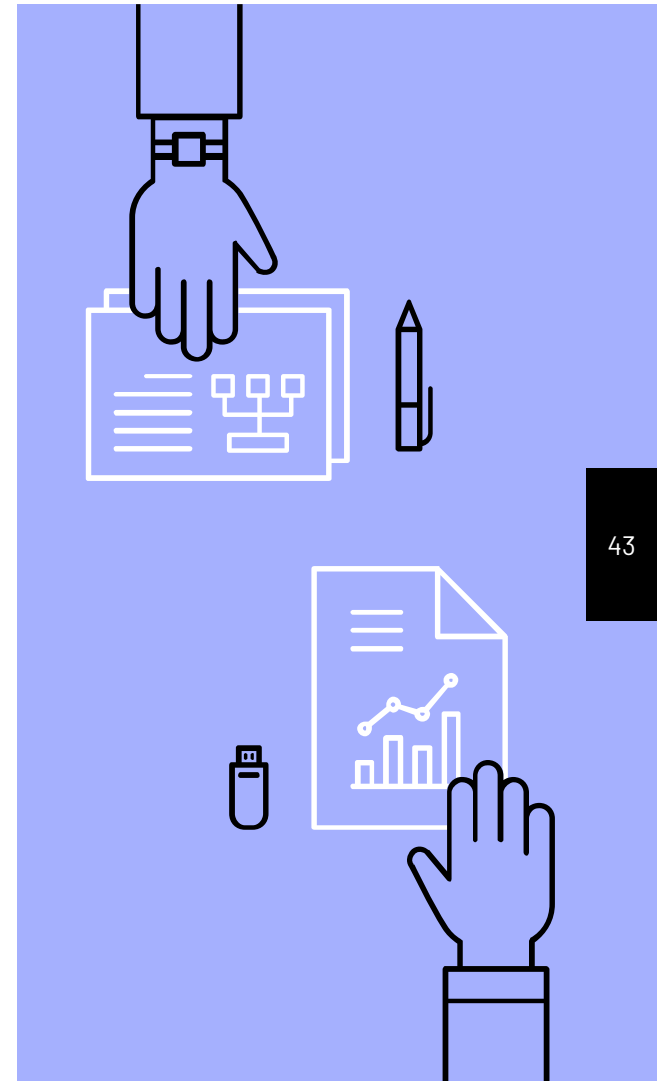
```
1 int IN3 = 5;    // Input3 conectada al pin 5
2 int IN4 = 4;    // Input4 conectada al pin 4
3 int ENB = 3;    // ENB conectada al pin 3 de Arduino
4 char leer;      // Variable para almacenar dato bluetooth
5
6 void setup() {
7     Serial.begin(9600); // Inia comunicación serial
8     pinMode (ENB, OUTPUT);
9     pinMode (IN3, OUTPUT);
10    pinMode (IN4, OUTPUT);
11
12 }
```

EJEMPLO 5: Comunicación inalámbrica bluetooth

```
14 void loop() {  
15     if (Serial.available() > 0) { // lee el bluetooth y almacena en estado  
16         leer = Serial.read();  
17     }  
18     //Preparamos la salida para que el motor gire en un sentido  
19     digitalWrite (IN3, HIGH);  
20     digitalWrite (IN4, LOW);  
21     // Si se recibe la letra 'a' motor con V 55  
22     if (leer == 'a') {  
23         // Aplicamos PWM al pin ENB, haciendo girar el motor, cada 2 seg aumenta la velocidad  
24         analogWrite(ENB, 55);  
25         Serial.print("Velocidad PWM: ");  
26         Serial.println(55);  
27     }  
28     // Si se recibe la letra 'b' motor con V 105  
29     else if (leer == 'b') {  
30         analogWrite(ENB, 105);  
31         Serial.print("Velocidad PWM: ");  
32         Serial.println(105);  
33     }  
34  
35     // Si se recibe la letra 'c' motor con V 255(Maxima)  
36     else if (leer == 'c') {  
37         analogWrite(ENB, 255);  
38         Serial.print("Velocidad Maxima: ");  
39         Serial.println(255);  
40     }  
41 }
```

¿QUÉ QUEDA POR APRENDER?

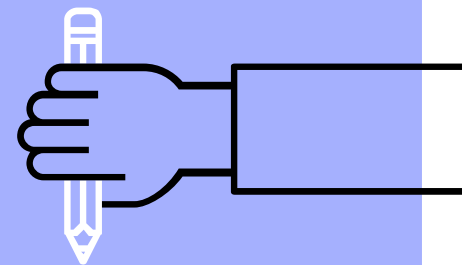
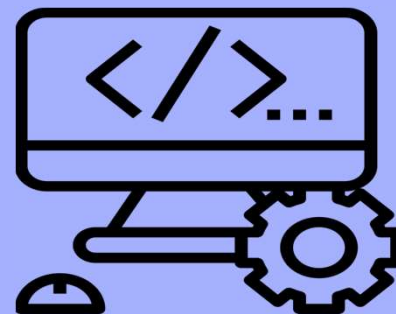
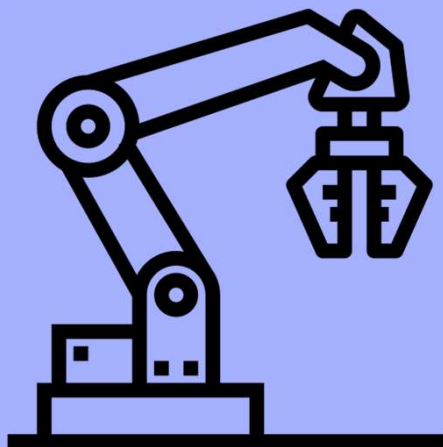
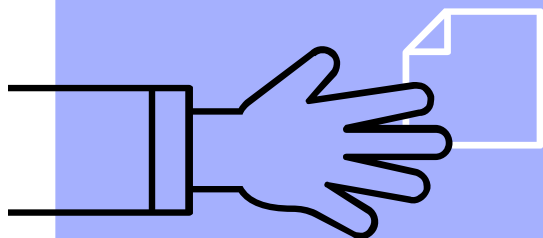
- ▶ Uso de periféricos: Shields, Pantallas, etc.
- ▶ Demás sensores
- ▶ Conexión a internet



PROYECTO: Robot (*OPCIONAL*)

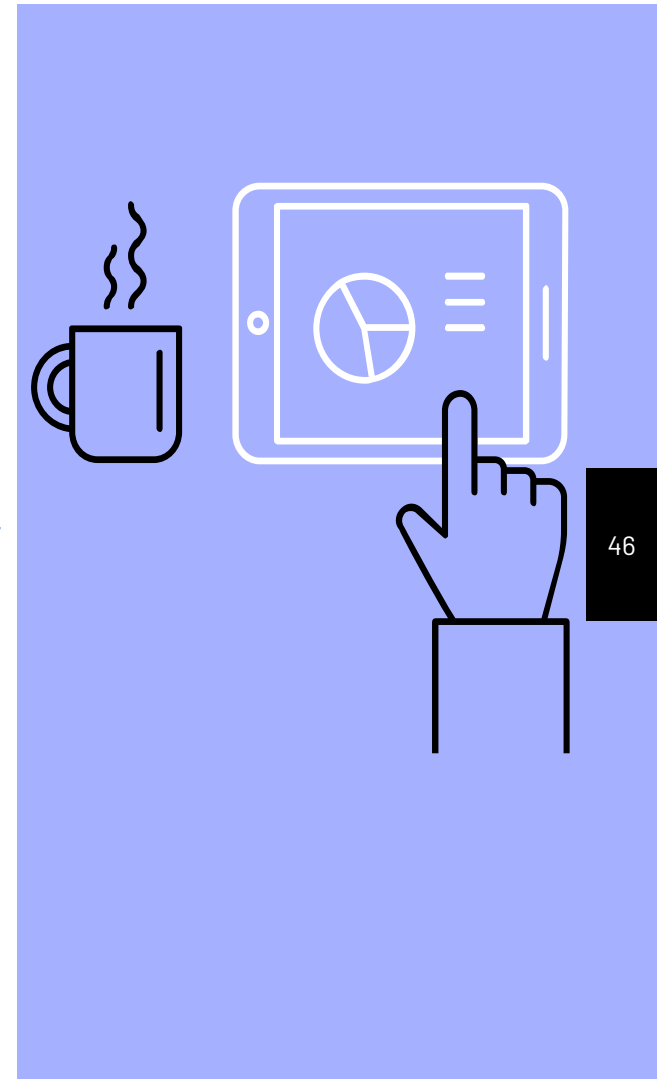


FIN!



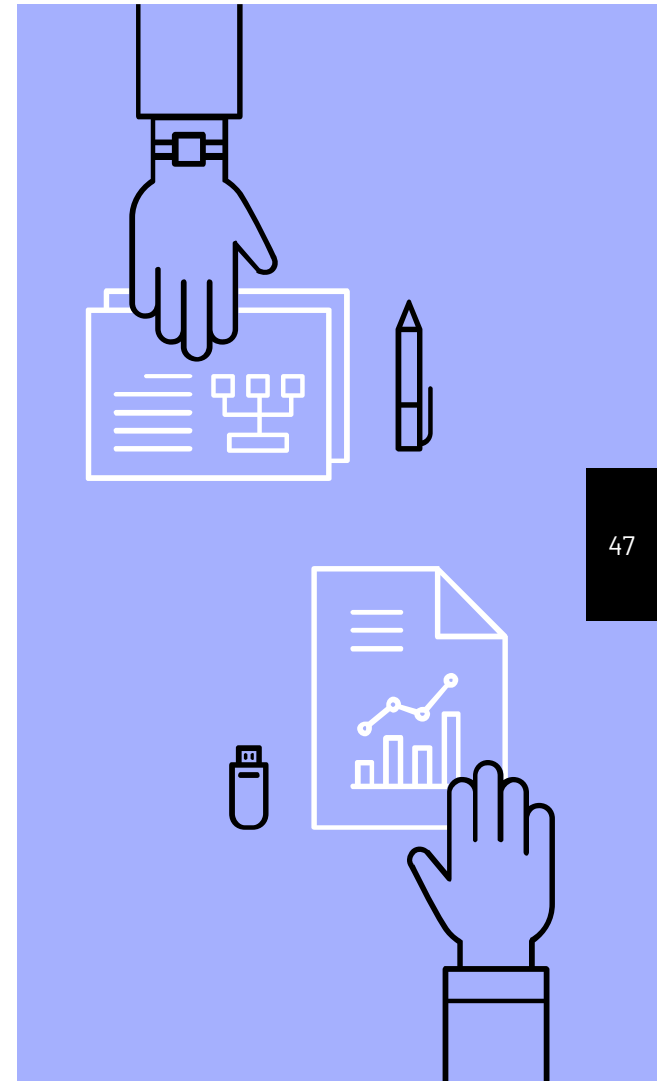
LISTA DE ENLACES

1. Referencia del lenguaje: www.arduino.cc/reference/en/
2. Imágenes libres: <https://es.freeimages.com/>
<https://www.freepng.es/> <https://thenounproject.com/>
www.flaticon.com
3. Blog sobre programación:
<https://programarfacil.com/blog/arduino-blog/curso-de-arduino/>
4. Tienda de materiales: <http://tdrobotica.co>
<https://www.vistronica.com/>
5. Libro kit básico TdRobótica:
https://issuu.com/tdrobotica/docs/libro_kit_basico/4
6. Medir distancia: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>



ANEXO: LISTA DE MATERIALES MÓDULO 2

- ▶ 1 Tarjeta Arduino con cable USB
- ▶ 1 sensor de temperatura LM35
- ▶ 1 Fotorresistencia (LDR)
- ▶ 1 motor DC 3-6V
- ▶ 1 módulo ultrasonido HC-SR04
- ▶ 1 Módulo Driver motor L298N
- ▶ 1 Módulo bluetooth HC-06 o HC-05
- ▶ 3 Leds 5mm (Cualquier color)
- ▶ 3 Resistencias (220-330) Ω
- ▶ 1 Resistencias 10k Ω
- ▶ 1 Potenciómetro (1-10k) Ω
- ▶ 10 Jumpers o más.
- ▶ 1 Protoboard

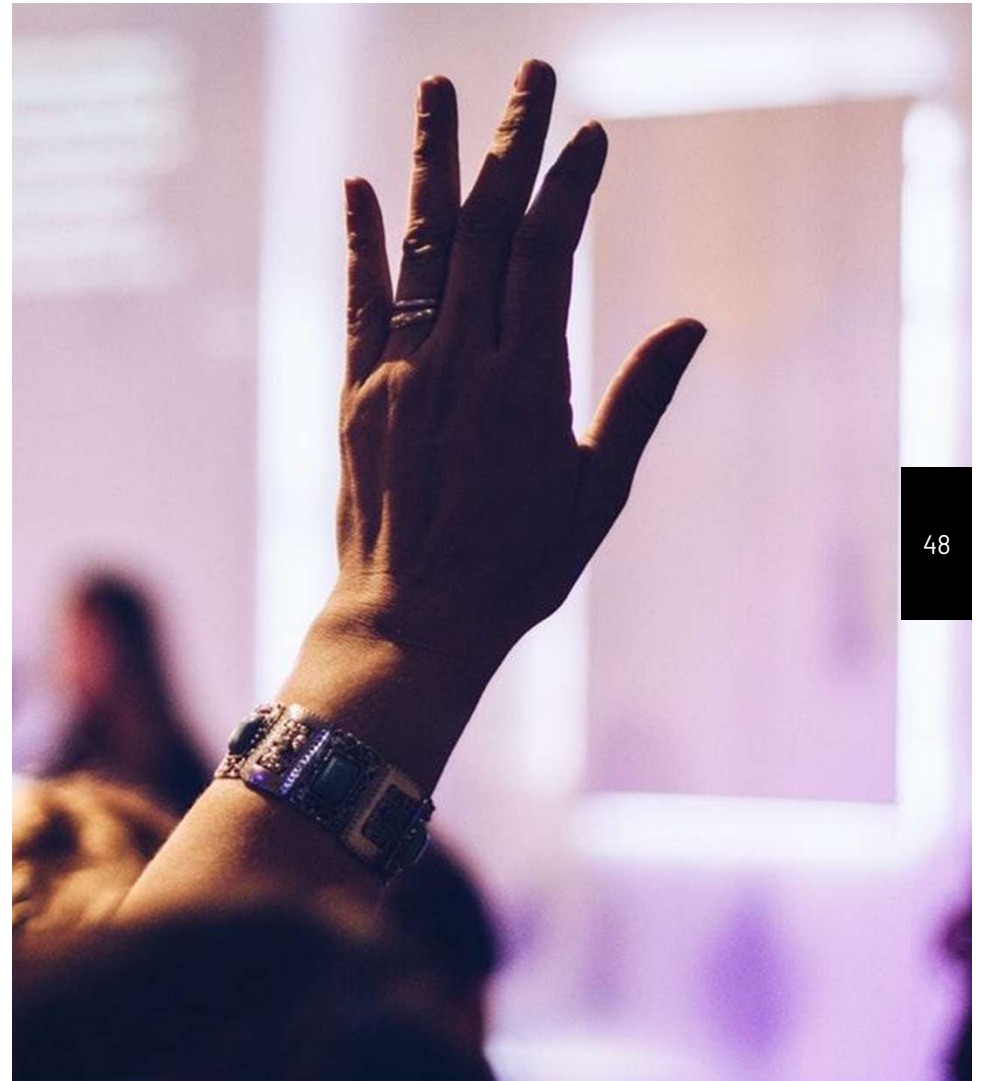


¿Alguna pregunta?

Contacto al correo UNAL:

juacastropa@unal.edu.co

GRACIAS!



CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- ▶ Presentation template by [SlidesCarnival](https://slidescarnival.com/)
- ▶ Photographs by [Unsplash](https://unsplash.com/)

