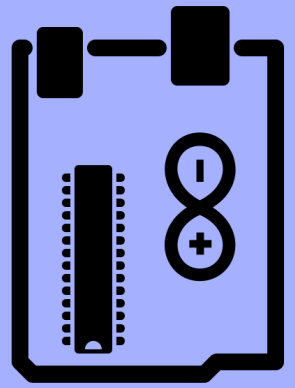
CURSO DE ARDUINO

MÓDULO 1

Por: Julián Andrés Castro
Estudiante ingeniería electrónica

CEIMTUN-RAS
UNIVERSIDAD NACIONAL DE COLOMBIA

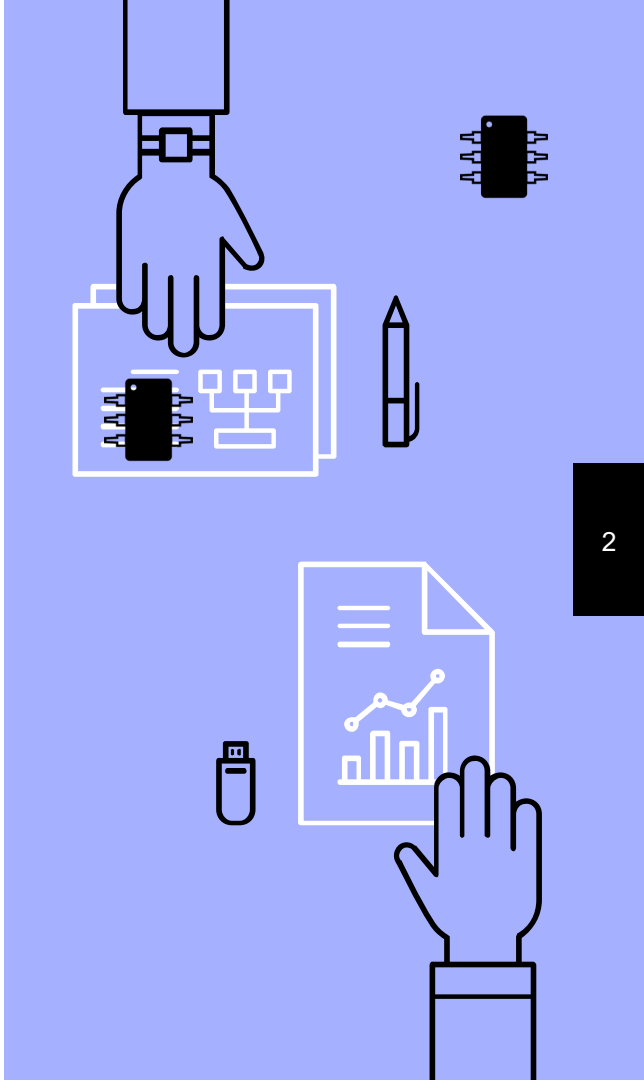


Conócenos:
Web: ceimtun

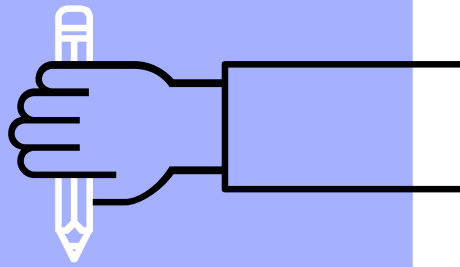
Redes:

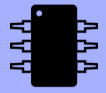
CONTENIDO: MÓDULO 1

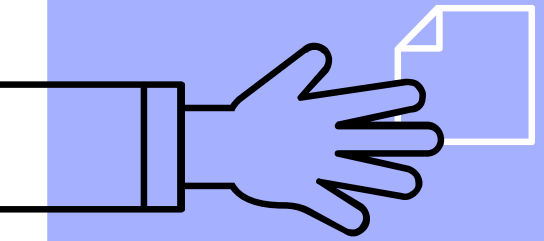
- Introducción: Computación física
- Software y hardware libre
- Arduino
- Referencias

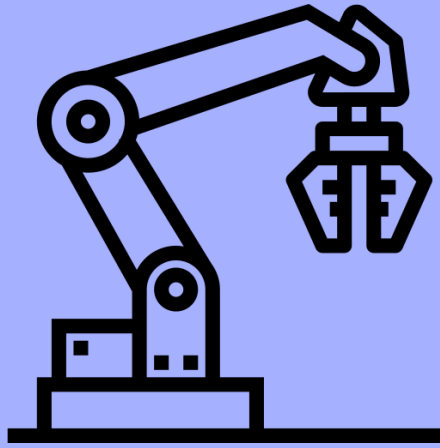


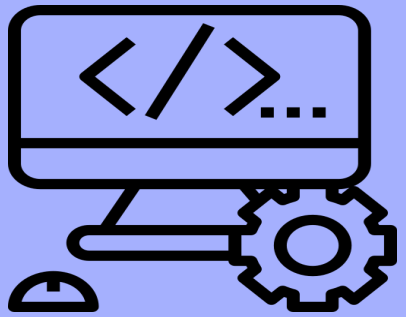
COMPUTACIÓN FÍSICA











COMPUTACIÓN FÍSICA

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

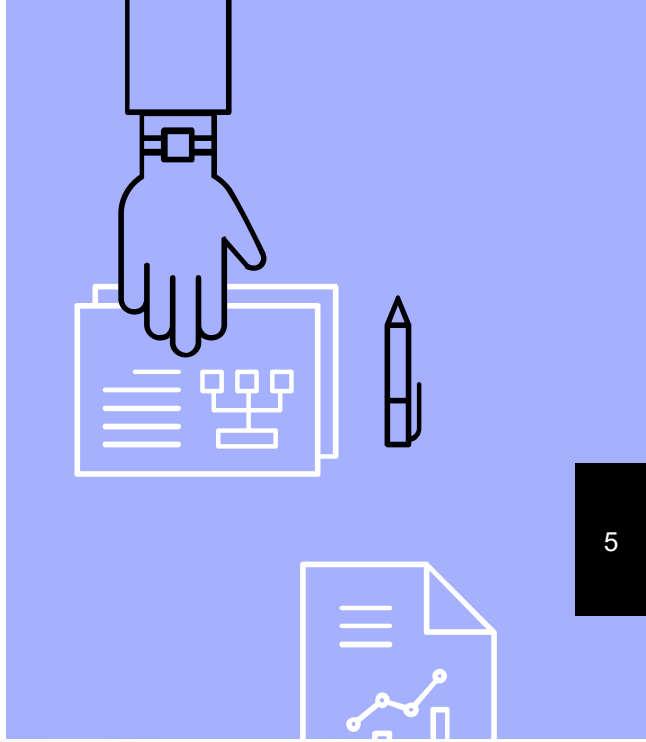
Transductores / Actuadores:

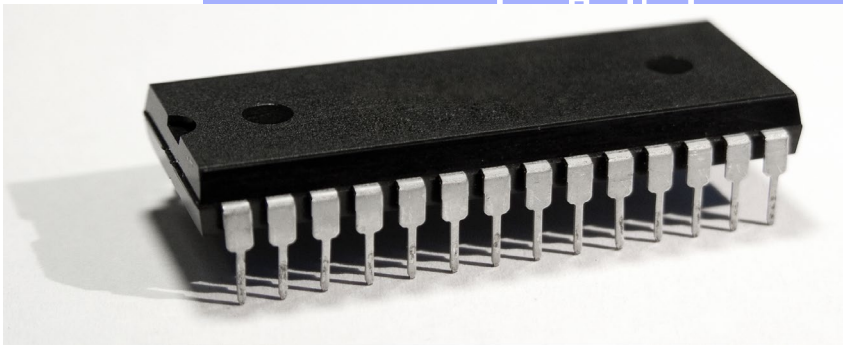
Transforman magnitudes físicas en señales eléctricas y viceversa.

MICROCONTROLADOR

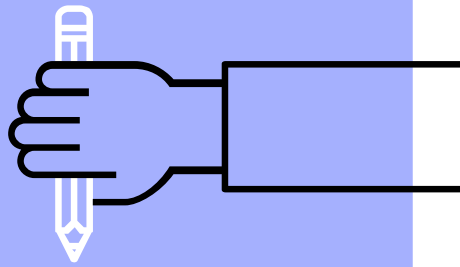
- Es un **circuito integrado programable**, capaz de ejecutar las órdenes que están almacenadas en su memoria.

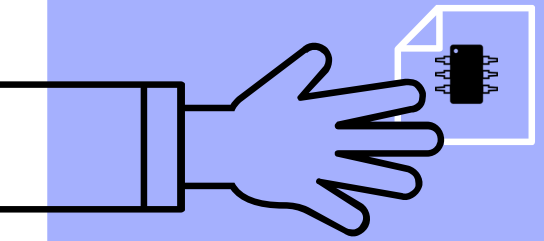
- También se llaman **MCU** por sus siglas en inglés *MicrocontrollerUnit*.





SOFTWARE Y HARDWARE LIBRE





GitHub

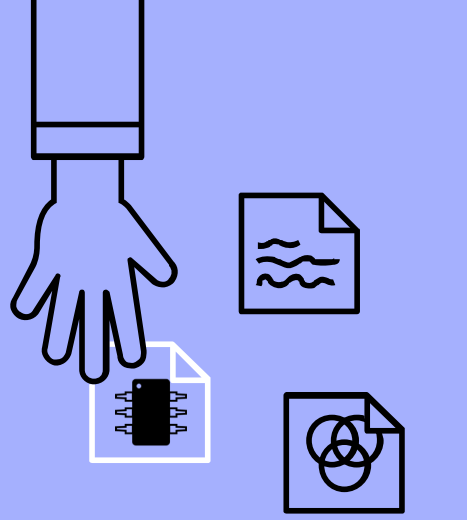
SOFTWARE Y HARDWARE LIBRE

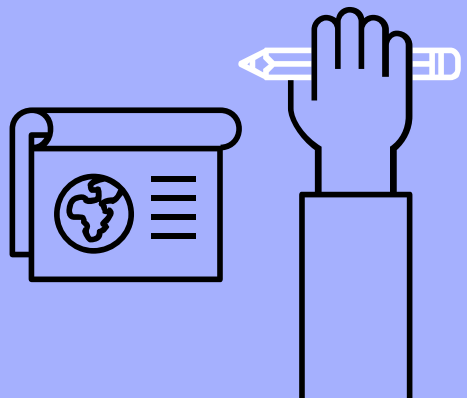
- SOFTWARE

Da la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

- HARDWARE

Dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público.

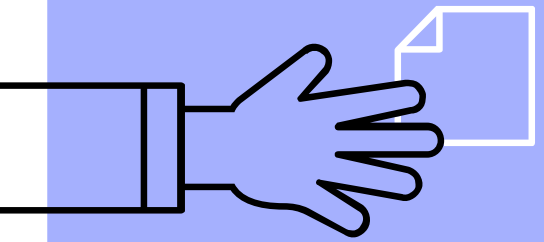


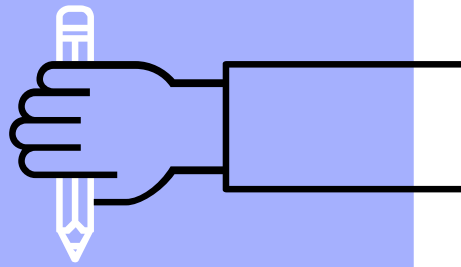


FREE SOFTWARE FOUNDATION

creative commons

ARDUINO

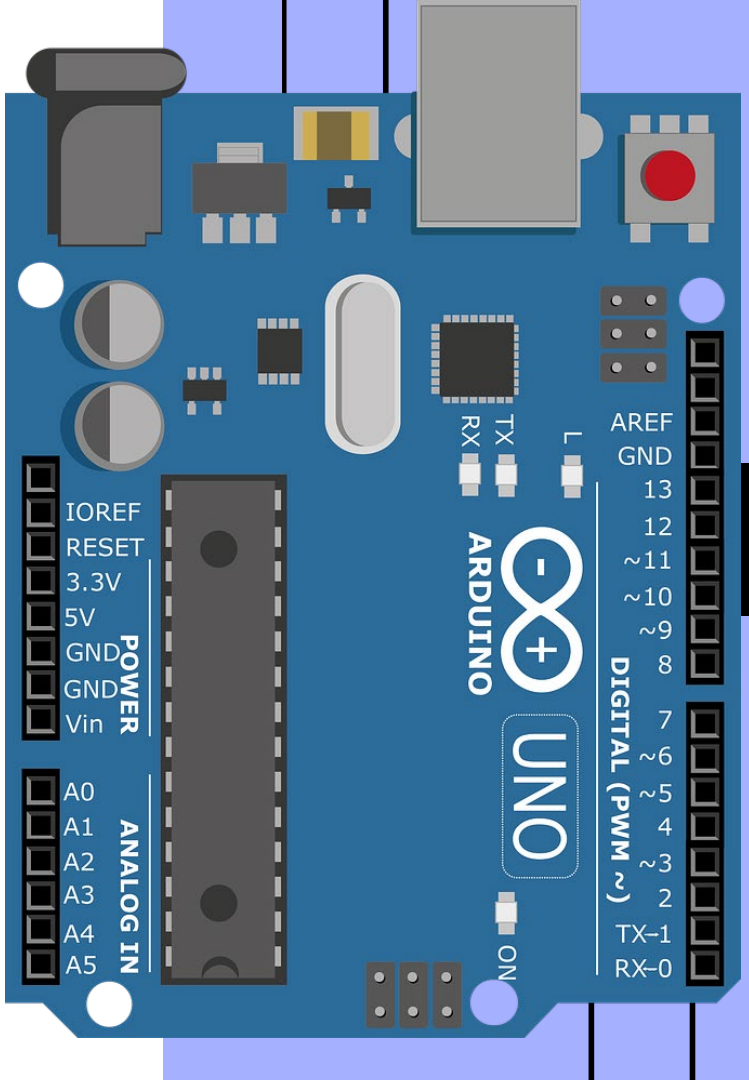




ARDUINO

ARDUINO

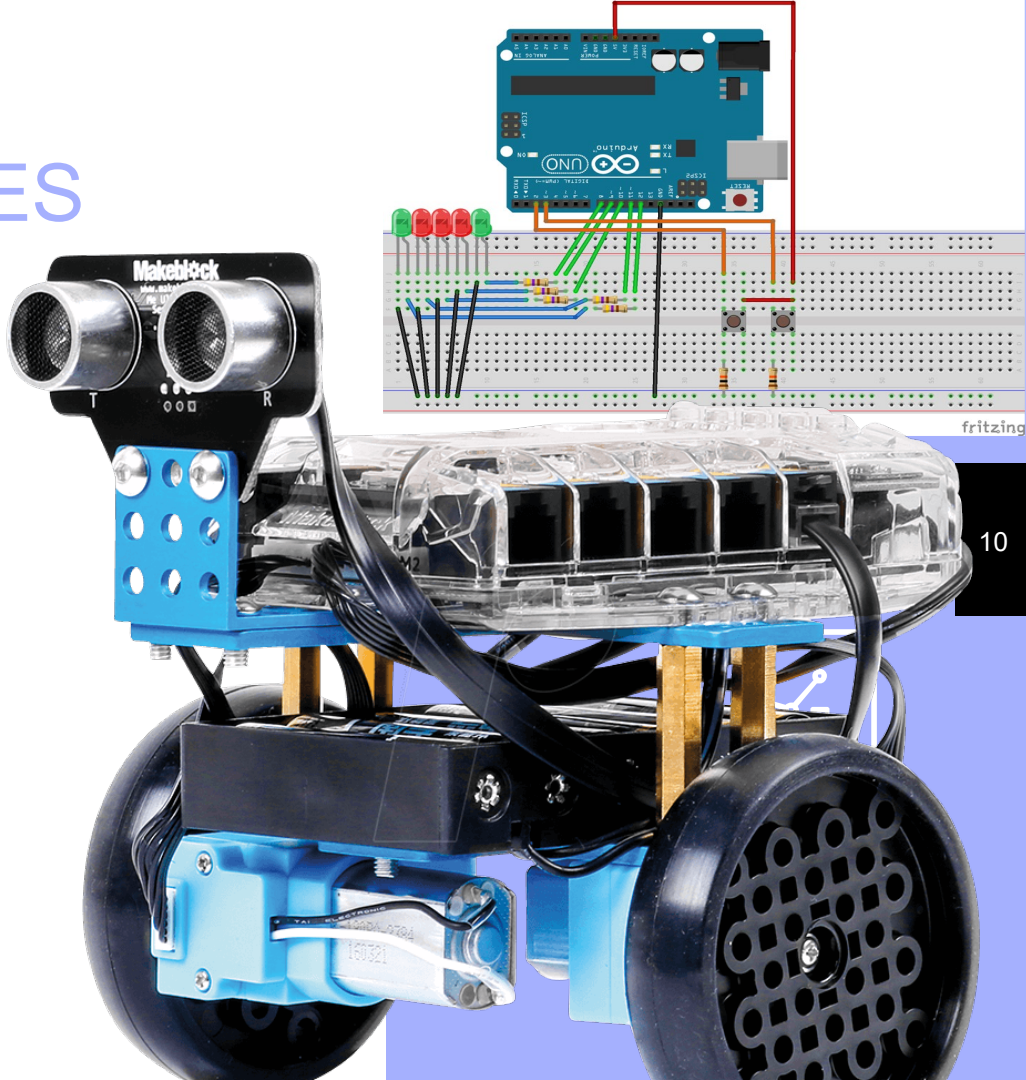
- Plataforma de electrónica abierta para la creación de prototipos.
- Es de bajo costo, flexible y fácil de usar.
- Creada para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos.
- Lenguaje basado en Wiring / procesing



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

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

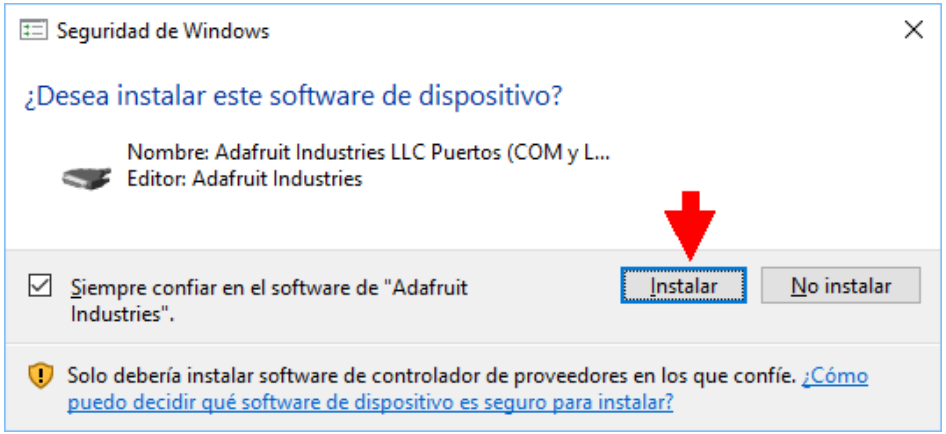
Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

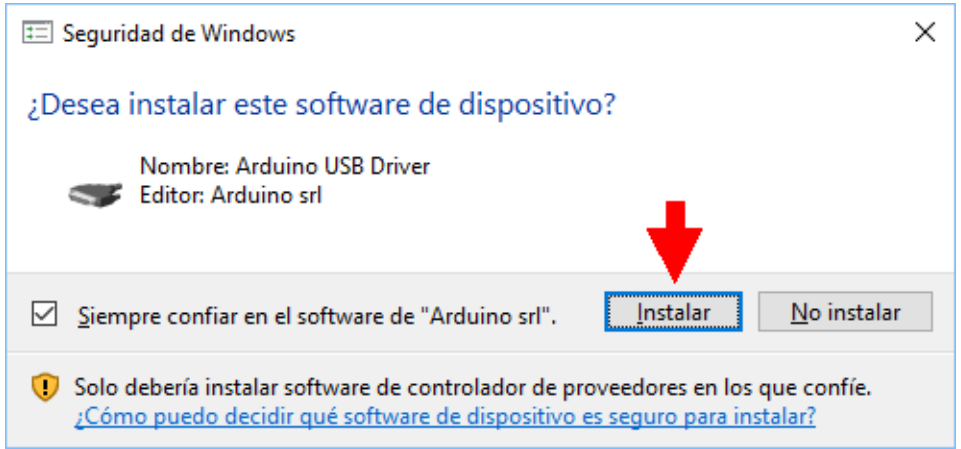
Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

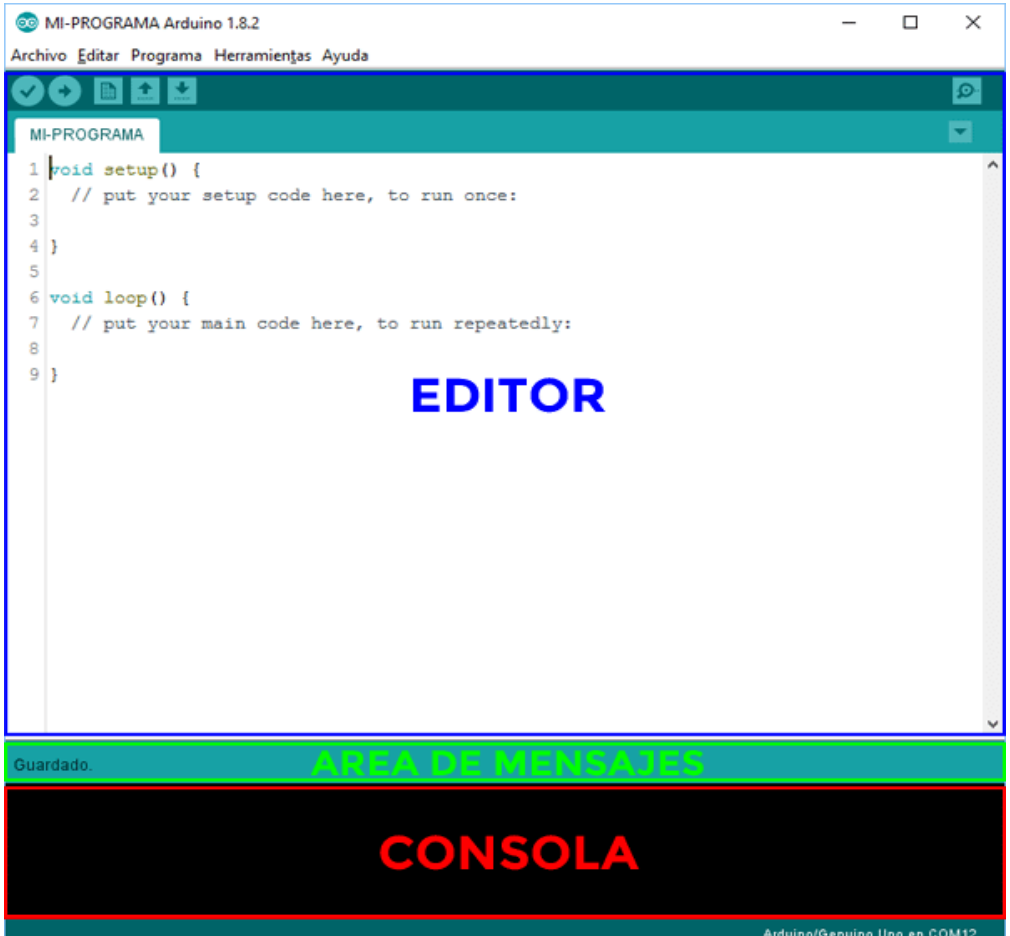
ENTORNO DE DESARROLLO



Instalar drivers

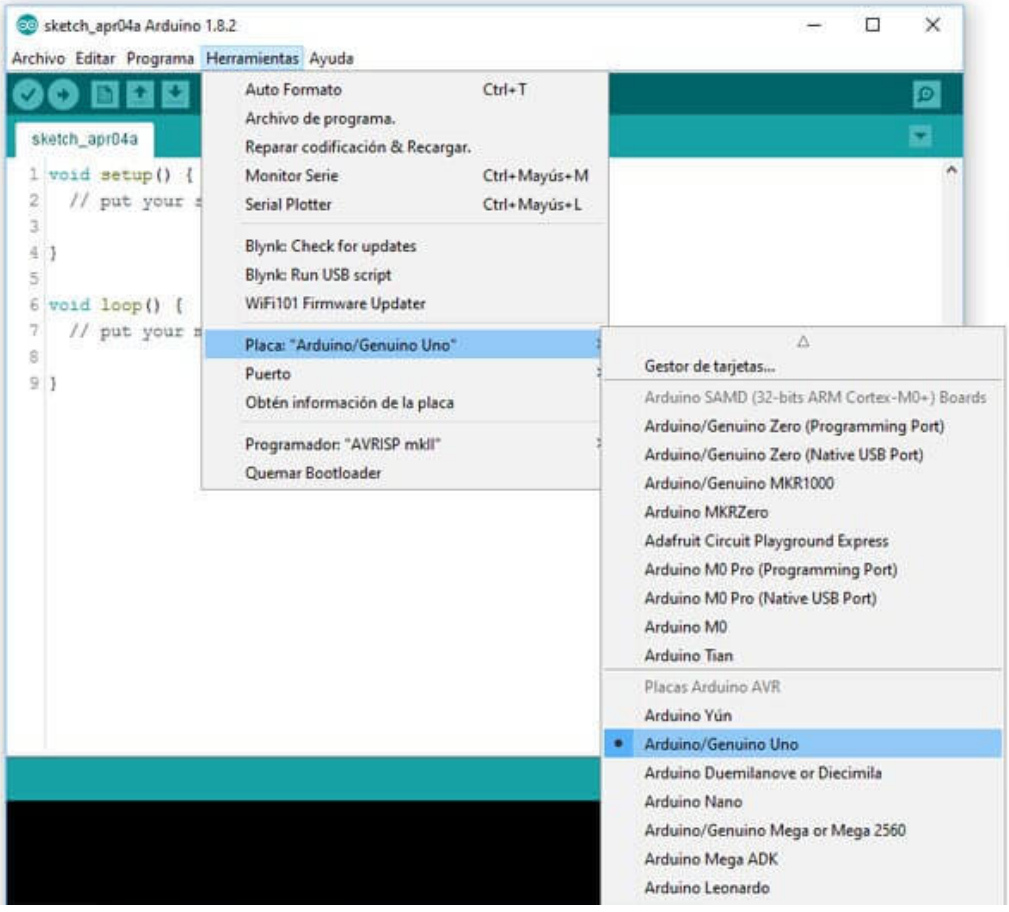


ENTORNO DE DESARROLLO IDE



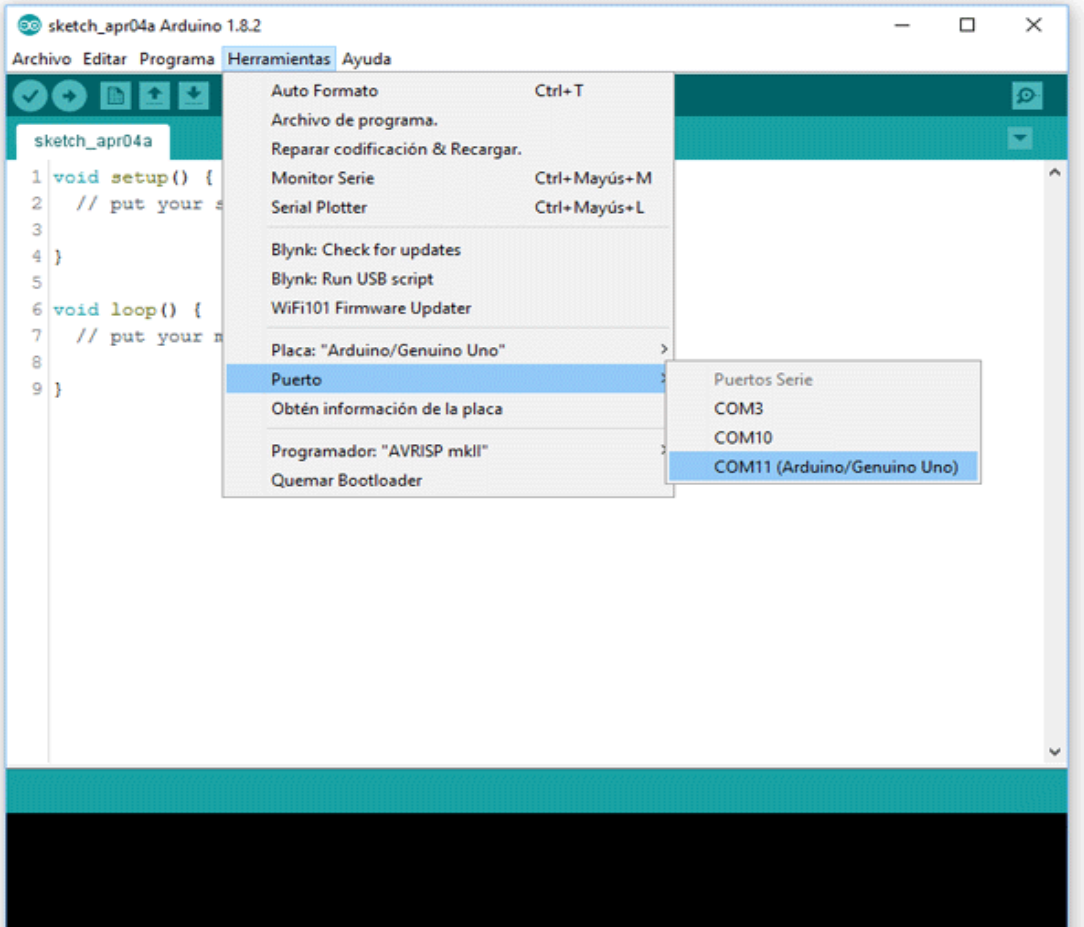
Partes fundamentales del IDE de Arduino

ENTORNO DE DESARROLLO



Seleccionar placa

ENTORNO DE DESARROLLO

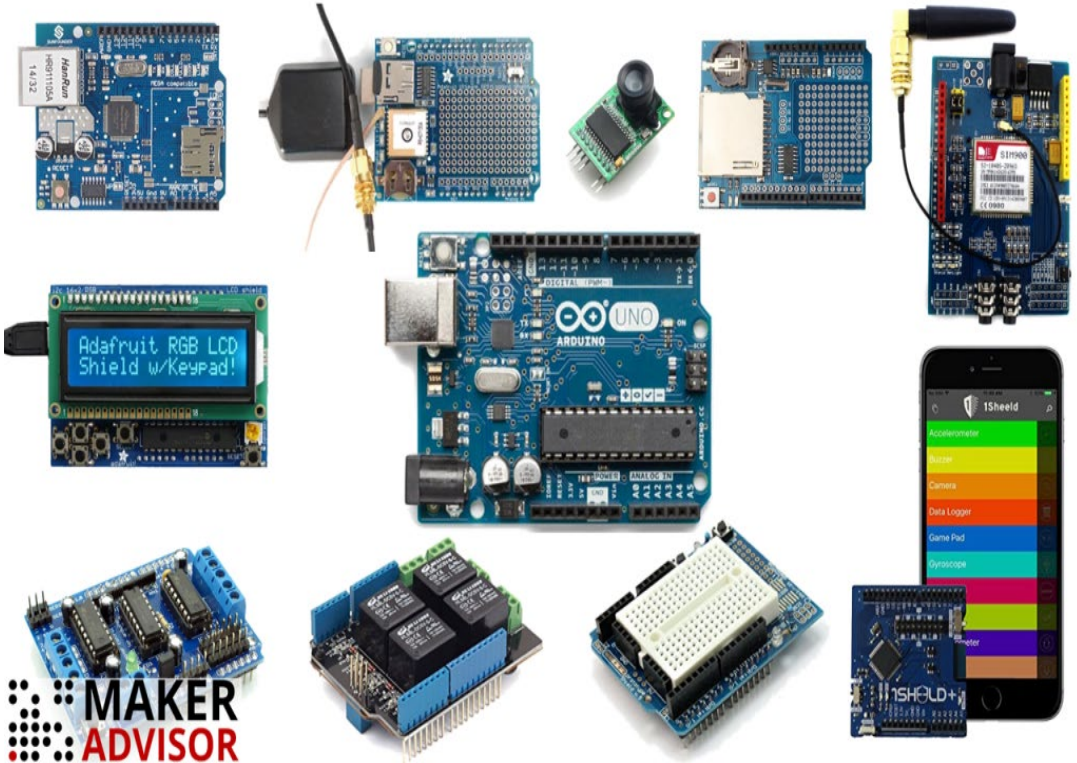


Seleccionar puerto

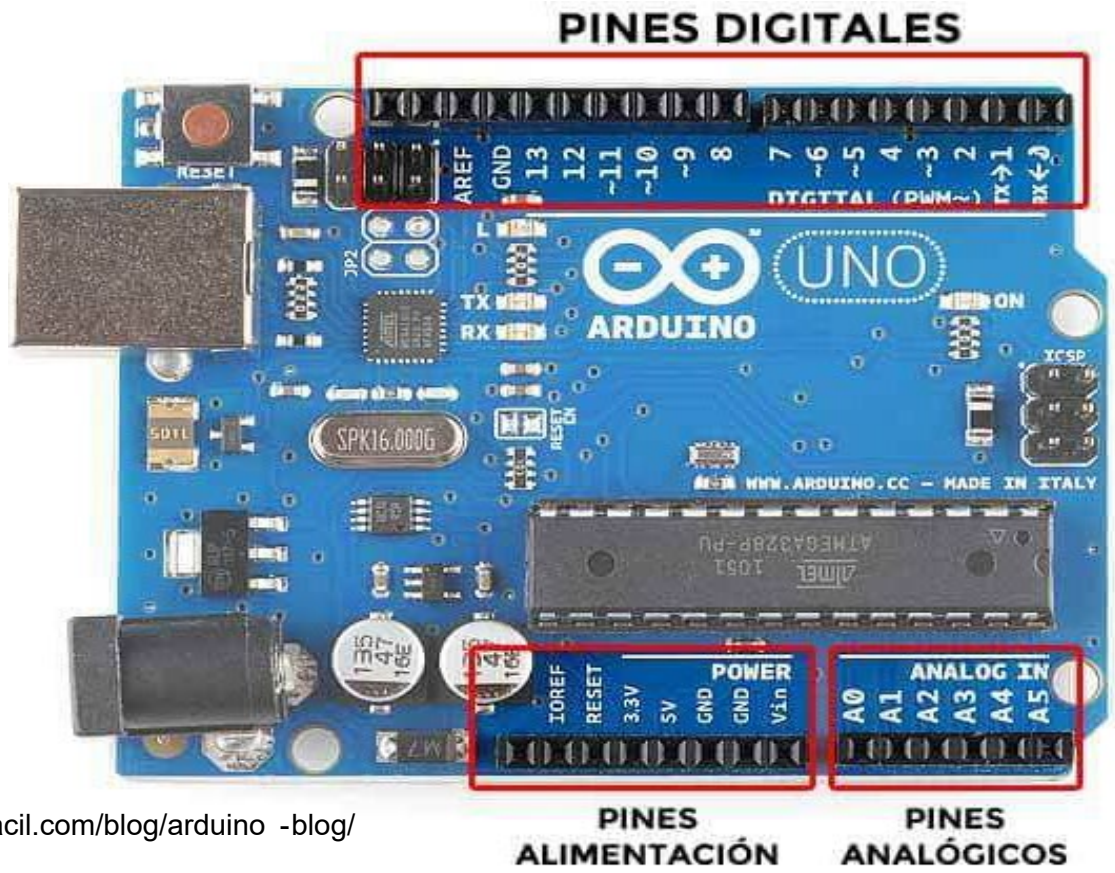
PLACAS DE ARDUINO

SHIELDS DE ARDUINO

Piezas de hardware que puede montar en el Arduino para darle un propósito específico o capacidades adicionales.



ARDUINO UNO



https://programarfacil.com/blog/arduino -blog/

ARDUINO UNO

PINES DIGITALES:

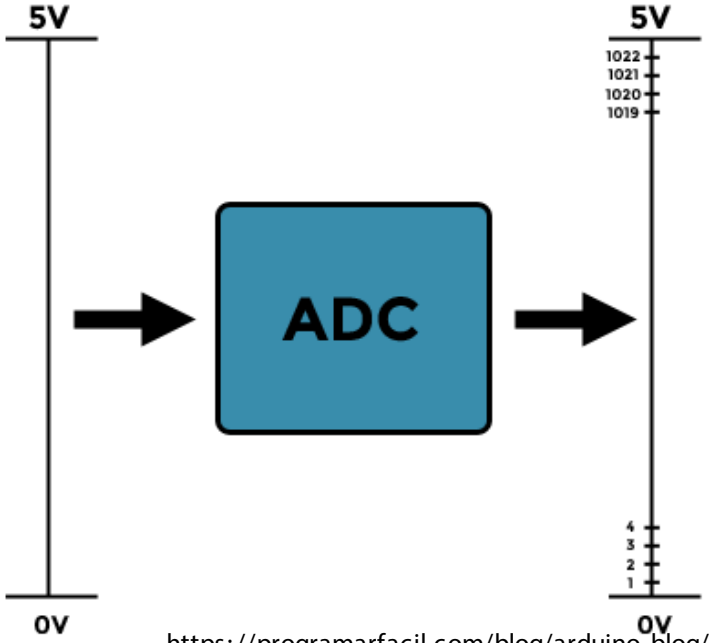
- Tiene 14 pines digitales que van del **0 al 13.**
- Dos estados **HIGH o LOW**
- Pines PWN *(pulse-width modulation)*

https://programarfacil.com/blog/arduino -blog/

ARDUINO UNO

PINES ANALÓGICOS:

- Medir diferentes voltajes entre 0 y 5 voltios.
- *AnalogDigital Converter con resolución de 10 bits.*
- *Rango de voltaje dividido en 1024 partes. (0-1023)*

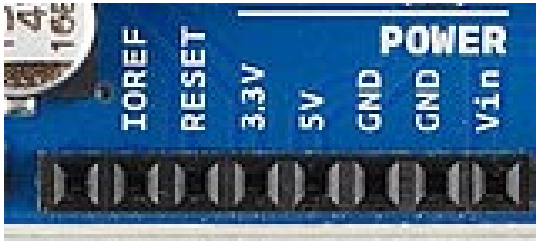


https://programarfacil.com/blog/arduino-blog/

ARDUINO UNO

PINES DE ALIMENTACIÓN:

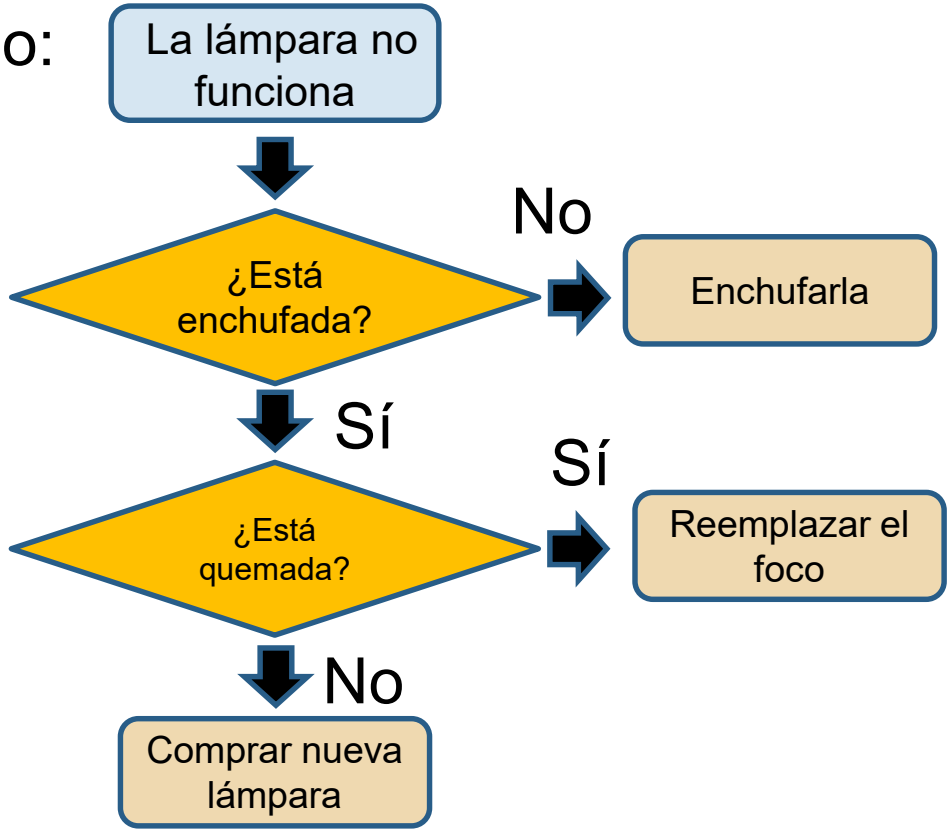
- 3,3V: suministra ese voltaje por ese pin.
- 5V: suministra ese voltaje por ese pin.
- GND: Es la toma de tierra y por donde debemos cerrar el circuito.



https://programarfacil.com/blog/arduino-blog/

LA PROGRAMACIÓN DE ARDUINO

Algoritmo:

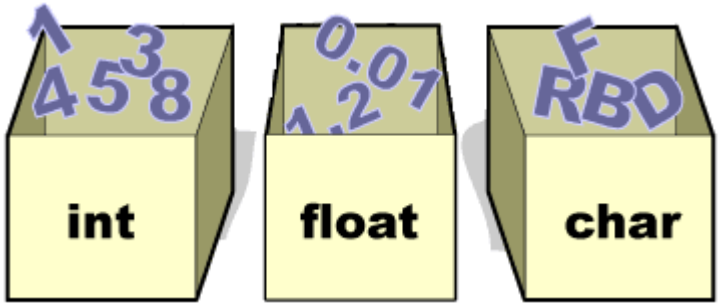


Lenguaje C++

Visitar:
www.arduino.cc/reference/en/

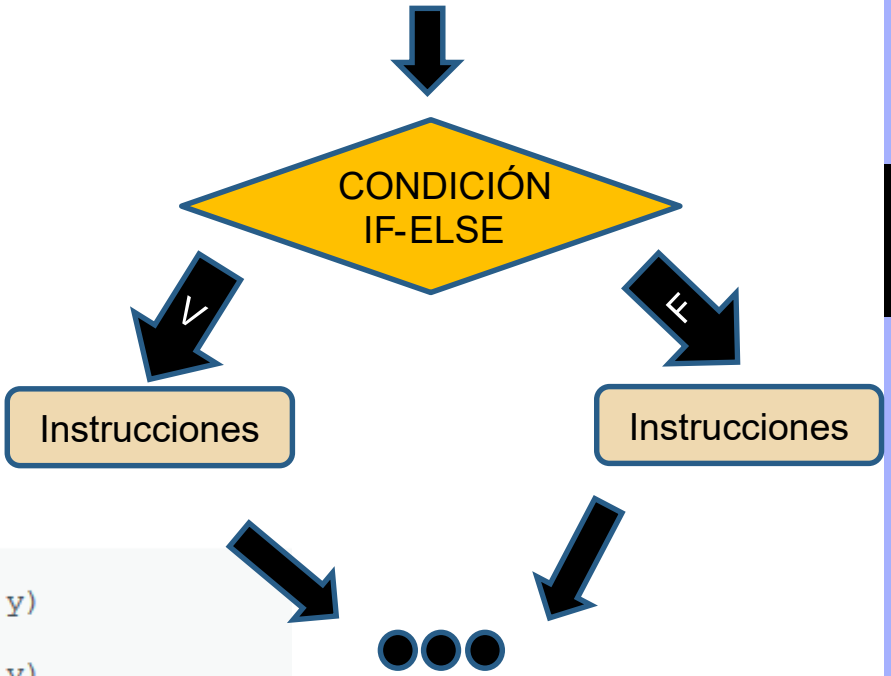
LA PROGRAMACIÓN DE ARDUINO

Tipos de variables



Tomado de : http://programacionmeta24.blogspot.com/

Condicionales:



Comparison Operators:

```
x == y (x is equal to y)
x != y (x is not equal to y)
x <  y (x is less than y)
x >  y (x is greater than y)
x <= y (x is less than or equal to y)
x >= y (x is greater than or equal to y)
```

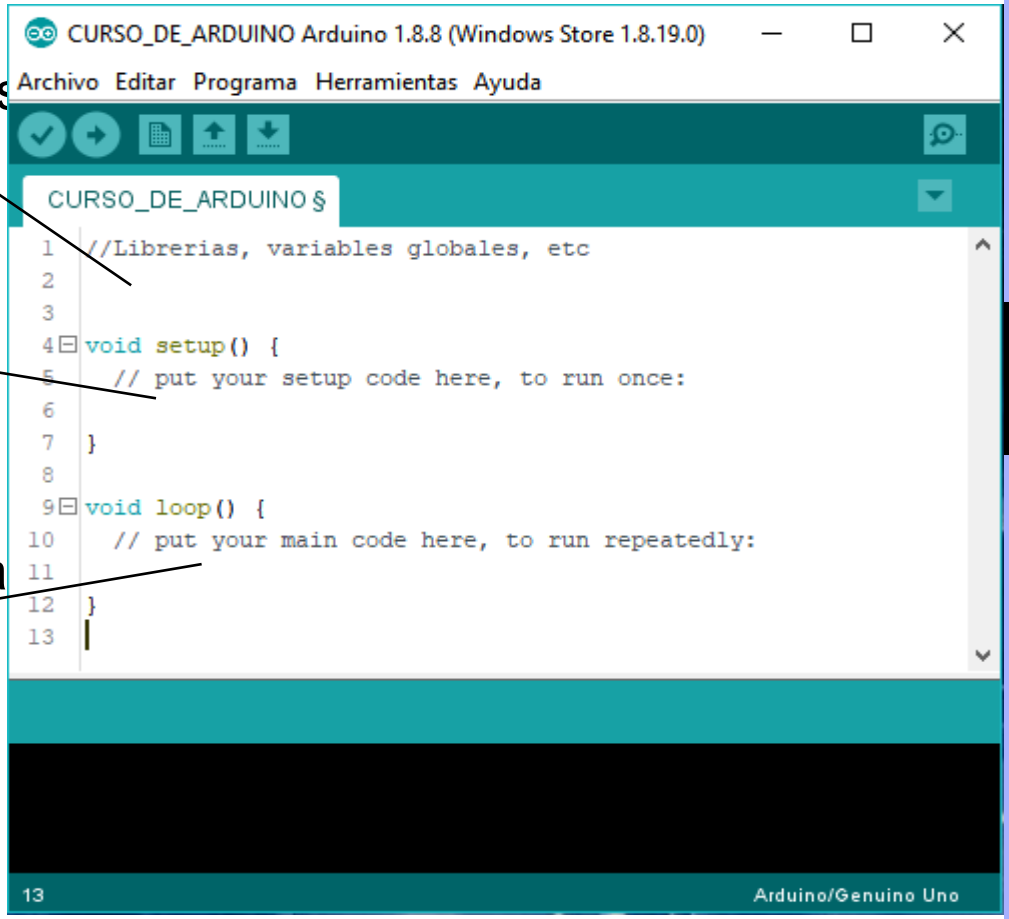
Estructura de un código - IDE

Declaración de variables y librerías

Se ejecuta al alimentar la placa.

Se ejecuta mientras se alimente la placa.

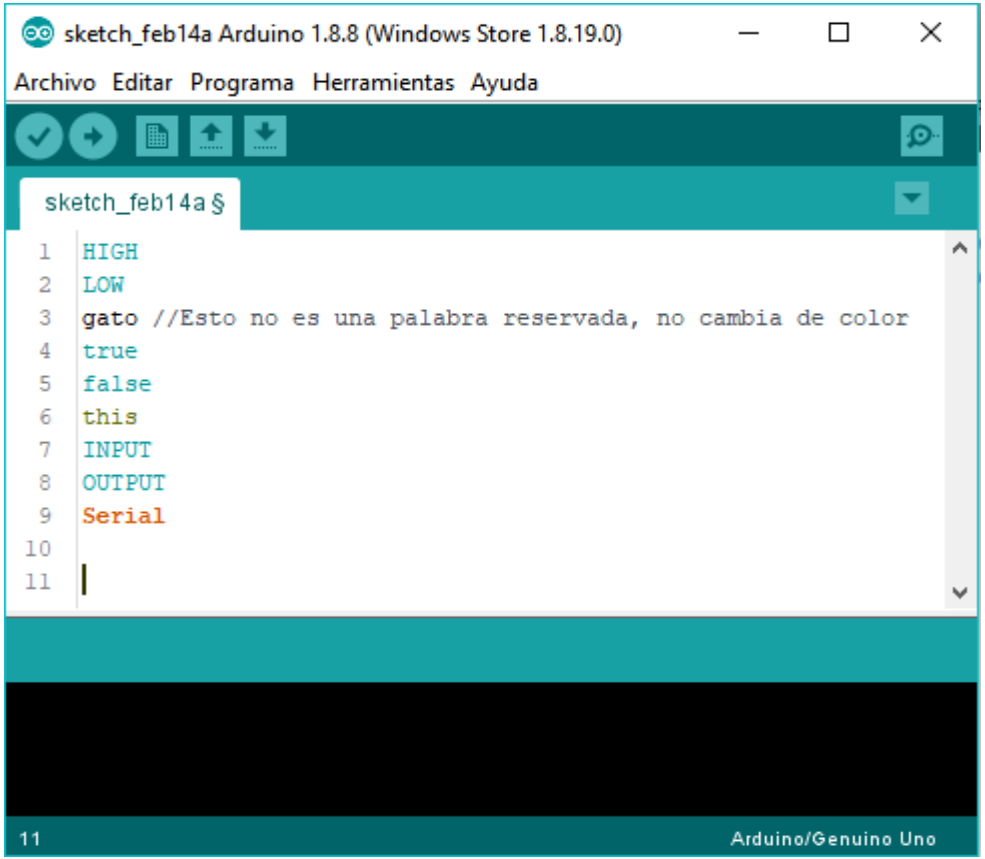
Consola



LA PROGRAMACIÓN DE ARDUINO

Palabras **reservadas** del **lenguaje** diferenciadas por **color**.

¡No las podemos usar!



Funciones principales en Arduino

Funciones digitales:

pinMode()

- Configurar un pin digital como salida o entrada.
- pinMode(# pin, mode)
- Ejemplo:

```
pinMode(13, OUTPUT);
pinMode(12, INPUT);
```

digitalRead()

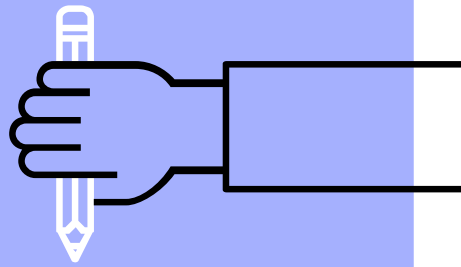
- Leer un pin digital (0 ó 1)
- digitalRead(# Pin)
- Ejemplo:

```
int a = digitalRead(13);
```

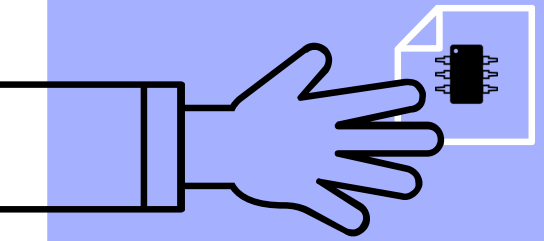
digitalWrite()

- Escribir un pin digital (0 ó 1)
- digitalWrite(# Pin, estado)
- Ejemplo:

```
digitalWrite (13, HIGH);
digitalWrite (13, LOW);
```



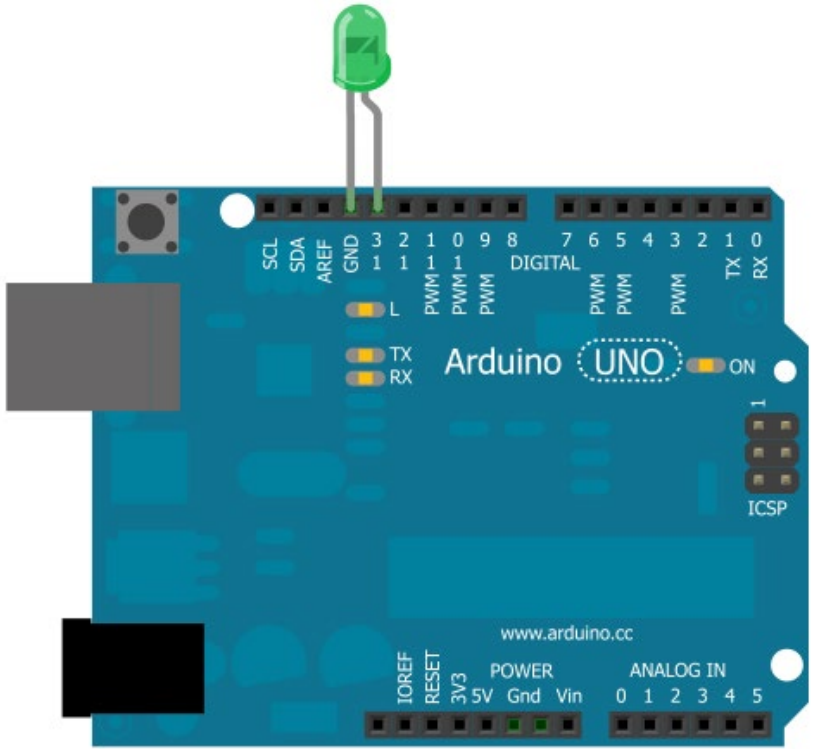
ACTIVIDADES



Descarga los Scripts en GitHub!

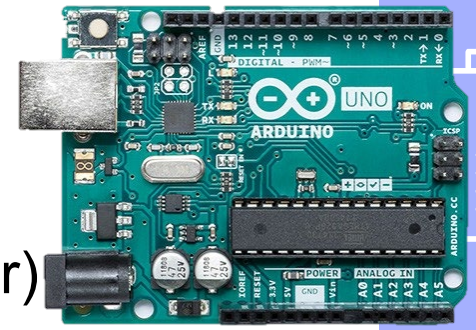
EJEMPLO 1: Blink - ¡Hola mundo!

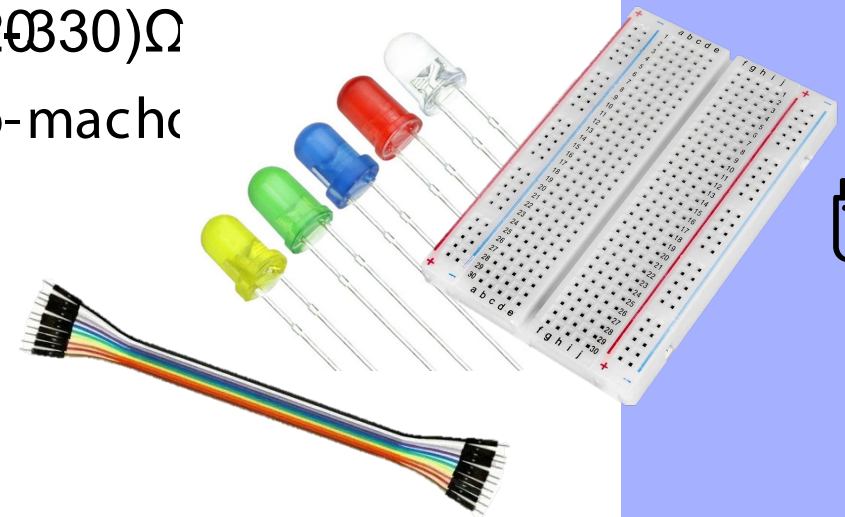
Encender y apagar un LED

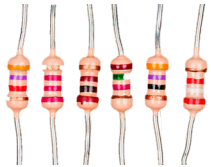


Materiales Ej. 1

- 1 Tarjeta Arduino
- 1 Led 5mm (cualquier color)
- 1 Resistencia (220-330)Ω
- 2 Jumpers macho-macho (cables)
- 1Protoboard







EJEMPLO 1 Blink - hola mundo!

Encender y apagar un LED

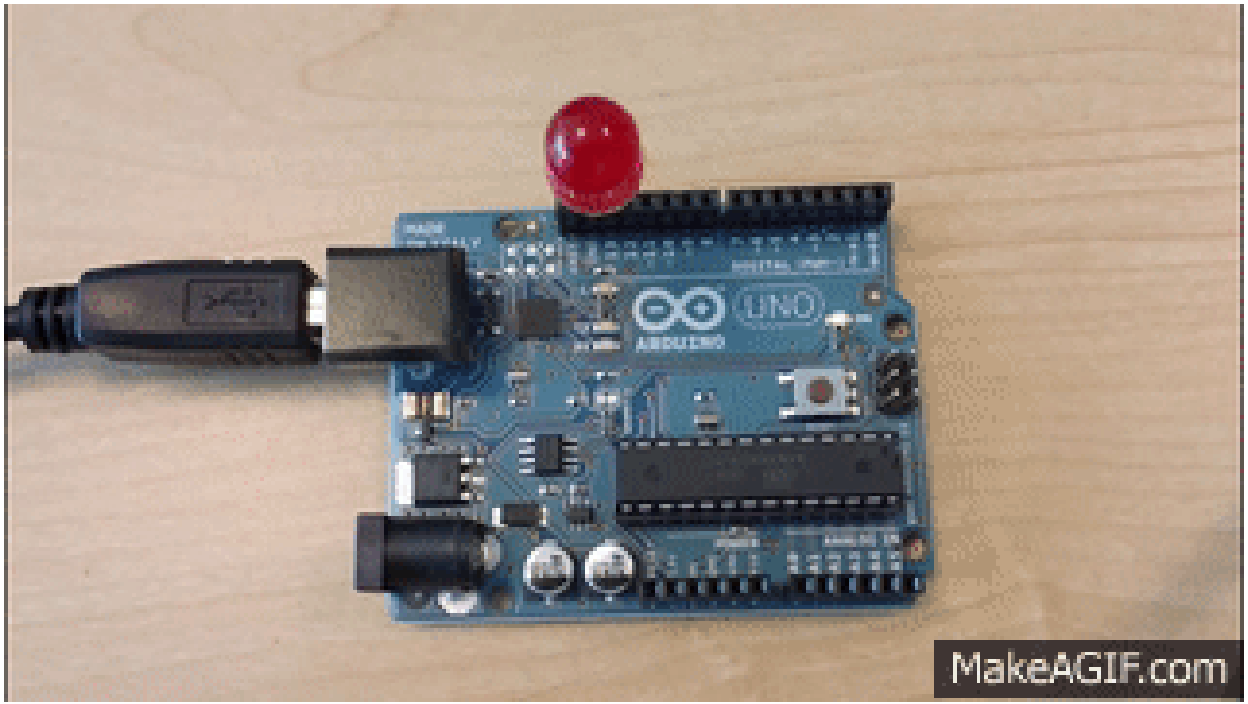
```
1  //El pin 13 esta conectado por defecto a una resistencia y un LED
2  int LED = 13;
3
```

```
4  // the setup function runs once when you press reset or power the board
5  void setup() {
6    // initialize digital pin LED as an output.
7    pinMode(LED, OUTPUT);
8  }
9
```

```
10 // the loop function runs over and over again forever
11 void loop() {
12   digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage level)
13   delay(1000);                       // wait for a second
14   digitalWrite(LED, LOW);    // turn the LED off by making the voltage LOW
15   delay(1000);                       // wait for a second
16 }
17
```

EJEMPLO 1: Blink - ¡Hola mundo!

Encender y apagar un LED:



Funciones principales en Arduino

Funciones análogas:

analogRead()

- Lee un valor análogo de 0 a 1023.
- analogRead(# pin)
- Ejemplo:

```
int a = analogRead(A0);
```

analogWrite() -> PWM

- Escribe un valor análogo de 0 a 255.
- analogWrite(Pin, ValorPWM)
- Ejemplo:

```
analogWrite(6, 124);
```

Estructuras

- Resumen de referencias del lenguaje para programar.

- setup()
- loop()

+Estructuras de control
- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

+Sintaxis
- ; (punto y coma)
- { } (llaves)
- / / (comentario de una sola línea)
- / * * / (comentario de varias líneas)
- # define
- # include

+Operadores matemáticos
- = (operador de asignación)
- + (suma)
- - (resta)
- * (multiplicación)
- / (división)
- % (módulo)

+Operadores de comparación
- == (igual que)
- ! = (diferente de)
- < (menor que)
- > (mayor que)
- <= (menor o igual a)
- >= (mayor o igual a)

+Operadores booleanos
- && (y)
- | | (o)
- ! (no)

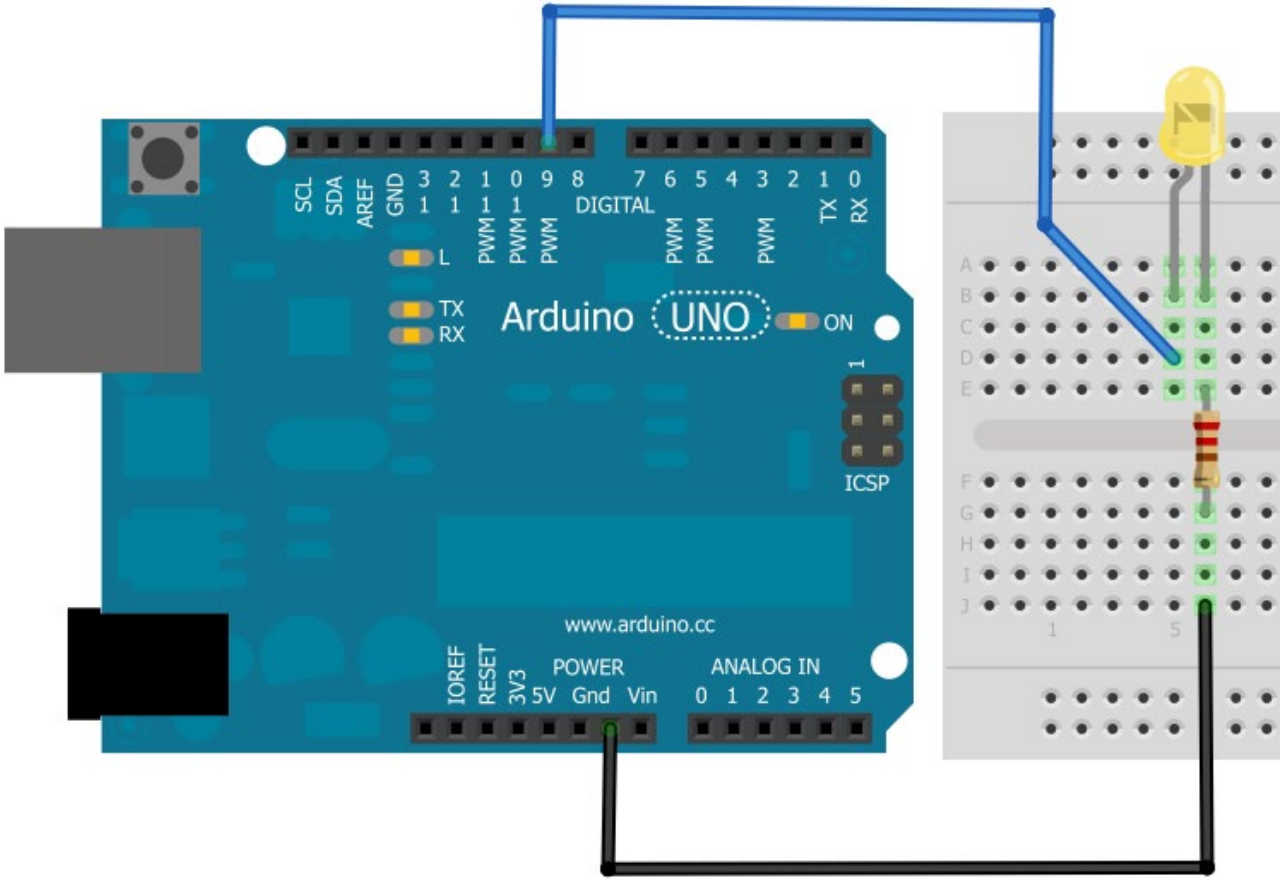
+Acceso con apuntadores
- * eliminar la referencia del operador
- & operador de referencia

+Operadores bit a bit
- & (bit a bit AND)
- | (bit a bit OR)
- ^ (bit a bit XOR)
- ~ (bit a bit NOT)
- << (a la izquierda BitShift)
- >> (a la derecha BitShift)

+Operadores compuestos
- + + (incremento)
- -- (decremento)
- + = (compuesto adición)
- - = (compuesto substracción)
- * = (compuesto multiplicación)
- / = (compuesto división)
-& = (compuesto bit a bit AND)
- | = (compuesto bit a bit OR)

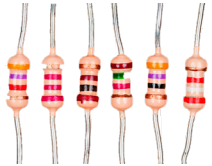
LanguageReference:https://www.arduino.cc/reference/en/

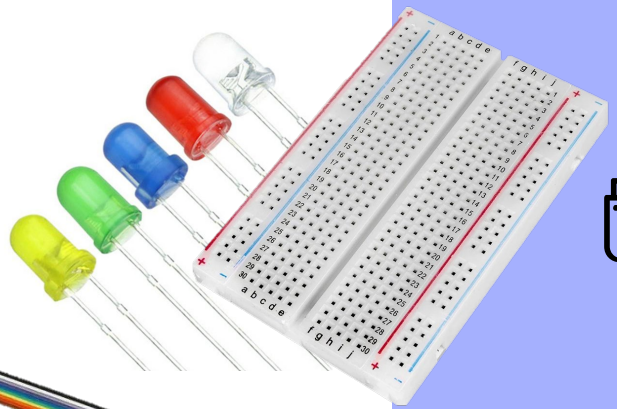
EJEMPLO 2: Control brillo PWM

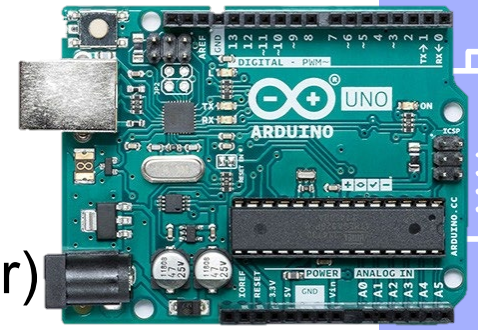


Materiales Ej. 2

- 1 Tarjeta Arduino
- 1 Led 5mm (cualquier color)
- 1 Resistencia (220-330)Ω
- 2 Jumpers macho-macho (cables)
- 1Protoboard







EJEMPLO 2: Control brillo PWM

```
/*
  ---------------------------------------------
  Enciende/Apaga un LED de forma proporcional
  ---------------------------------------------
*/
//-----------------------------------------------------
//Declara puertos de entradas y salidas y variables
//-----------------------------------------------------
int brillo = 0; //Variable de brillo inicia en 0
int variacion = 5; //Variable de incremento configurada de 5 en 5
int led = 9; //Pin donde se encuentra el LED, salida

//-------------------------------------
//Función principal
//-------------------------------------
void setup () {// Se ejecuta cada vez que el Arduino se inicia
  pinMode(led, OUTPUT); //Configurar el LED como una salida
}

```

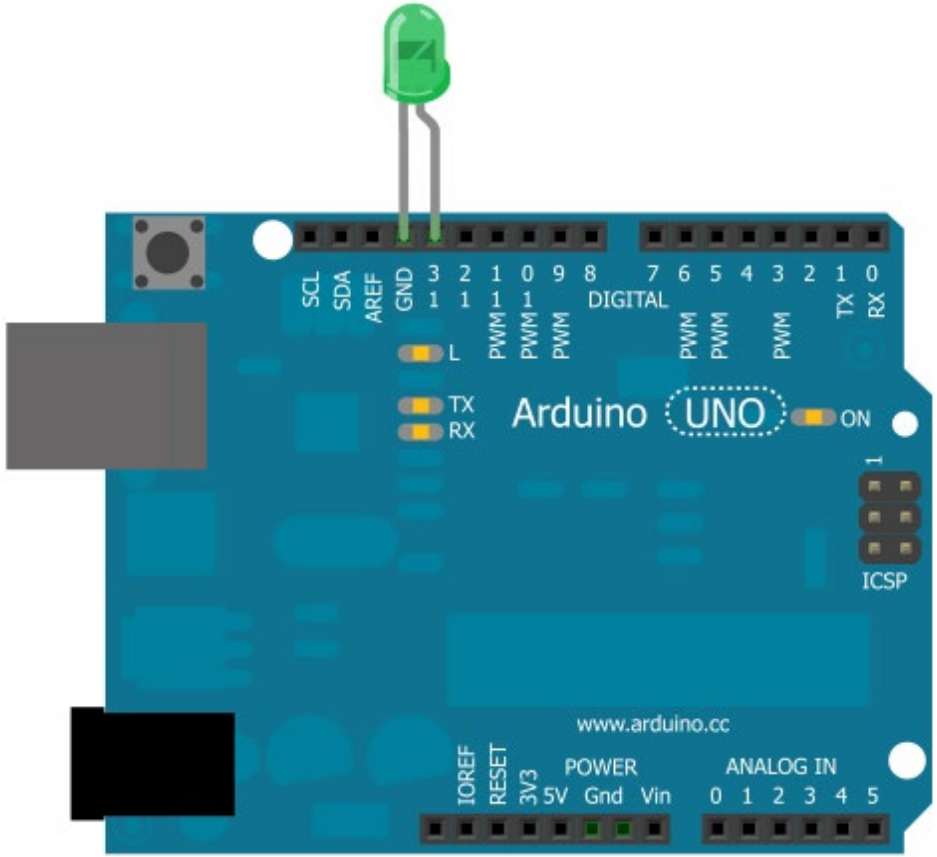
EJEMPLO 2: Control brillo PWM

```
20  //-------------------------------------
21  //Funcion cíclicla
22  //-------------------------------------
23  void loop () { // Esta función se mantiene ejecutando
24
25    // Escritura analoga (PWM) en el LED escribo el valor de brillo
26    analogWrite(led, brillo);
27    // Incremento la variable brillo de 5 en 5
28    brillo = brillo + variacion;
29    // Nota: PWM ----> 0 - 255
30    // Si el brillo es 0 o 255
31    if (brillo == 0 || brillo == 255)
32      variacion = -variacion; //La variación se vuelve negativa
33    delay (30); //Tiempo de incremento en el brillo
34  }
35  //Fin programa
36
```

EJEMPLO 3: Comunicación serial

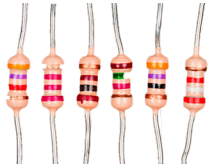
- Protocolo para comunicación entre dispositivos.

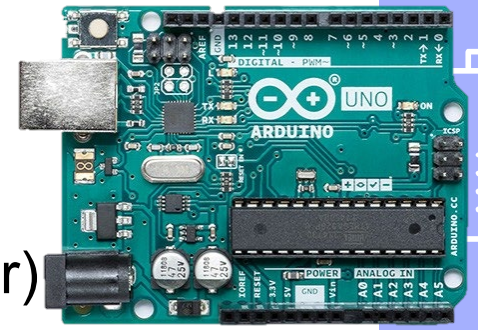
- Serial.begin(# baudios)
- Serial.println("string")
- Serial.read()
- Serial. … ()

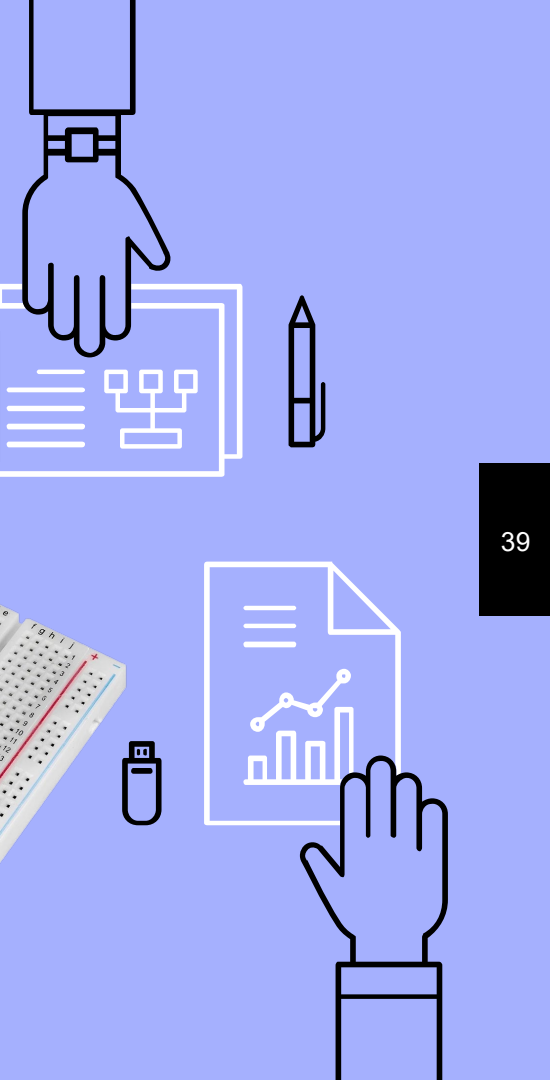


Materiales Ej. 3

- 1 Tarjeta Arduino
- 1 Led 5mm (cualquier color)
- 1 Resistencia (220-330)Ω
- 2 Jumpers macho-macho (cables)
- 1Protoboard







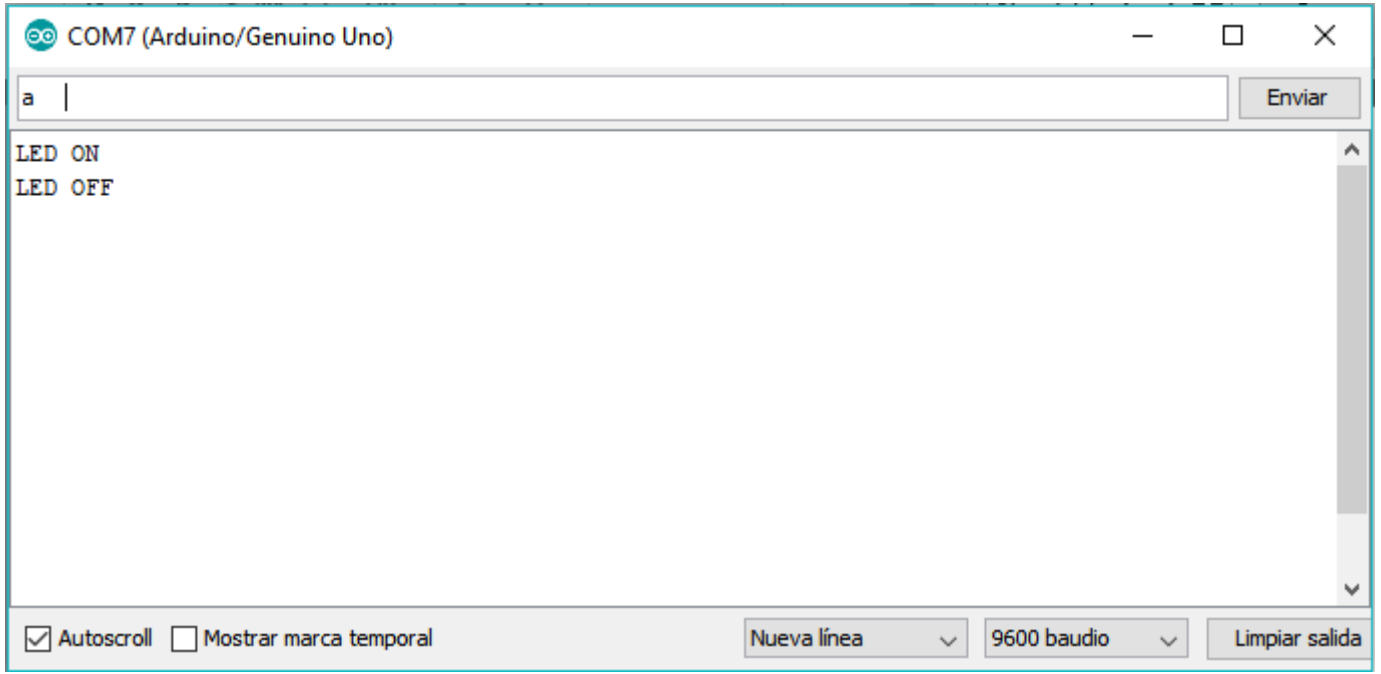
EJEMPLO 3: Comunicación serial

```
/*
  ------------------------------------
  ESCRITURA SERIAL
  ------------------------------------
*/

//-----------------------------------------------------
//Declara puertos de entradas y salidas y variables
//-----------------------------------------------------
int led = 13;   //Pin donde se encuentra el LED, salida
char leer;      //Variable donde se almacena la letra
boolean encendido = false; //Estado LED la primera vez, apagado


//-----------------------------------
//Función principal
//-----------------------------------
void setup(){// Se ejecuta cada vez que el Arduino se inicia

  Serial.begin(9600); //Inicia comunicación serial
  pinMode(led, OUTPUT); //Configurar el LED como una salida
}

```

EJEMPLO 3: Comunicación serial

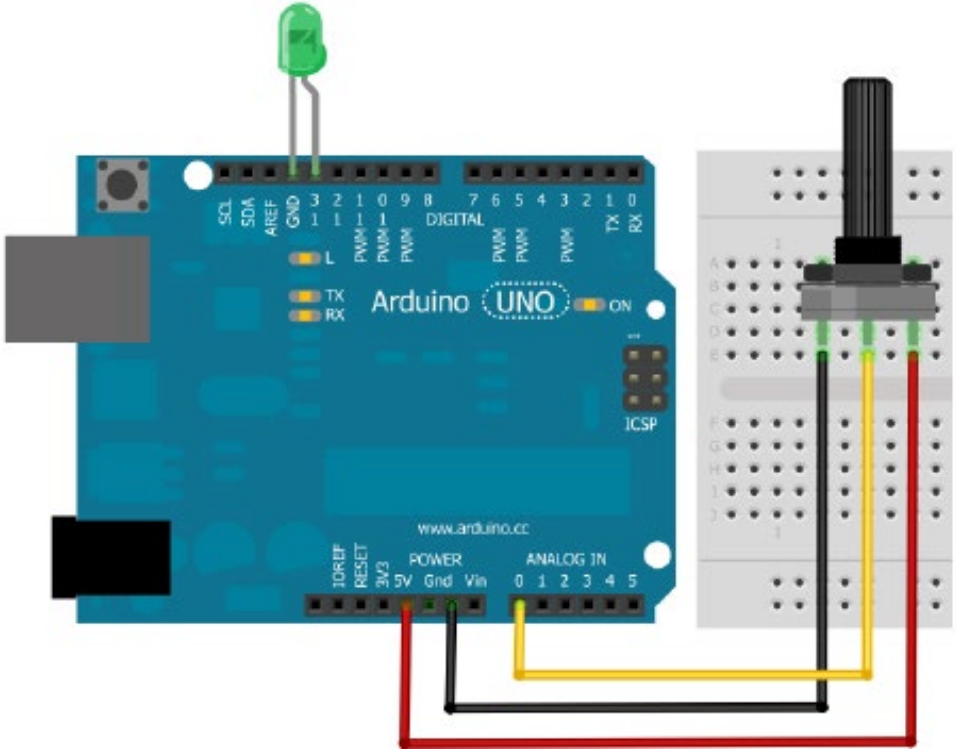
```
24  //-----------------------------------
25  //Funcion cíclicla
26  //-----------------------------------
27  void loop(){ // Esta funcion se mantiene ejecutando
28
29    //Guardar en una variable el valor de la consola serial
30    leer = Serial.read();
31
32    // Si es la letra 'a' y además el LED está apagado
33    if ( (leer == 'a') && (encendido == false) ) {
34      digitalWrite(led, HIGH); // Enciende el LED
35      Serial.println("LED ON"); //Escribe en pantalla el texto
36      encendido = true; // Actualiza el estado del LED
37    }
38    // Si es la letra 'a' y además el LED está encendido
39    else if ( (leer == 'a') && (encendido == true) ) {
40      digitalWrite(led, LOW); // Apaga el LED
41      Serial.println("LED OFF"); //Escribe en pantalla el texto
42      encendido = false; // Actualiza el estado del LED
43    }
44  }
45  //Fin programa
46
```

EJEMPLO 3: Comunicación serial



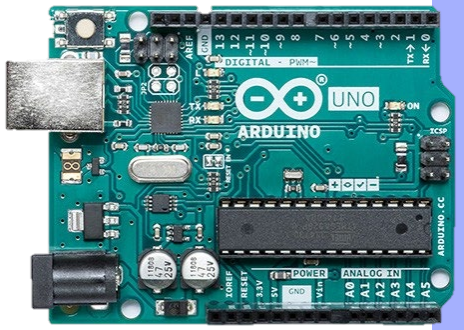
EJEMPLO 4: Lectura Analógica

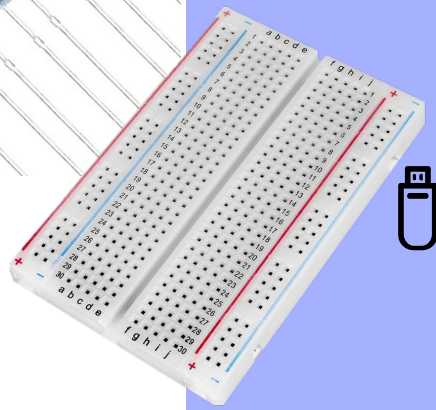
Control ON/OFF potenciómetro

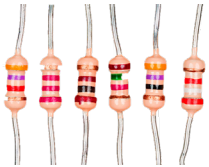


Materiales Ej. 4

- 1 Tarjeta Arduino
- 1 Led 5mm (cualquier color`
- 1 Resistencia (220-330)Ω
- 3 Jumpers macho-macho (cables)
- 1Potenciómetro (1-10)kΩ
- 1Protoboard







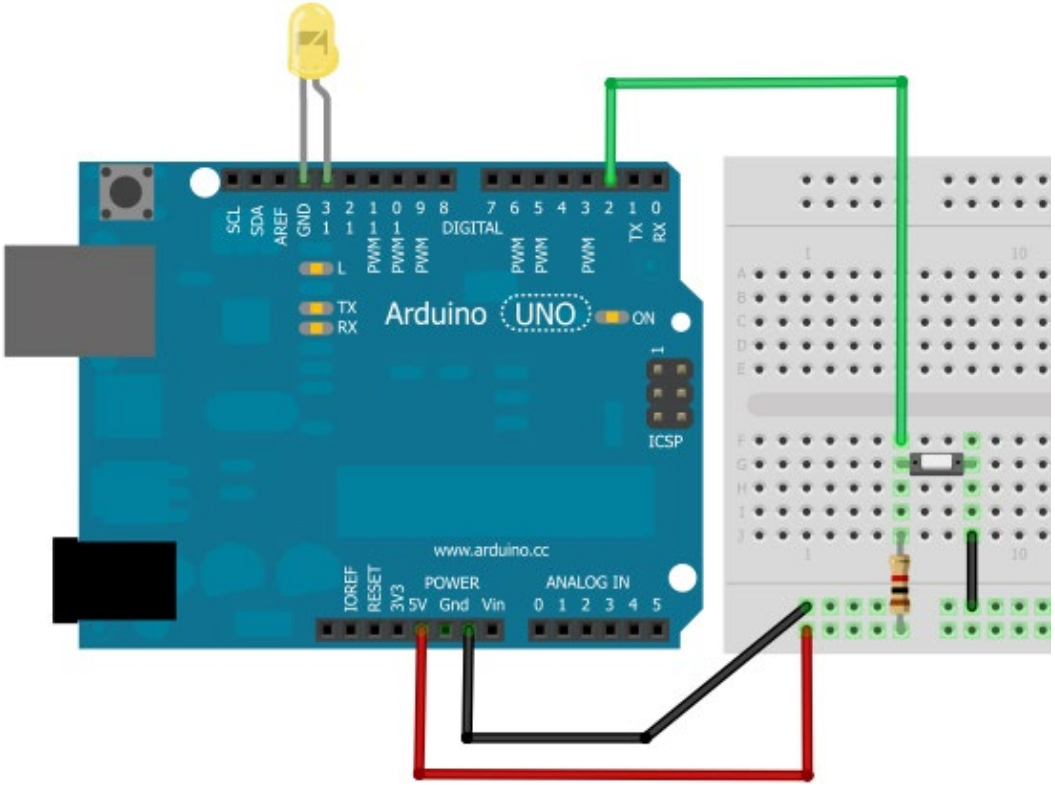
EJEMPLO 4: Control ON/OFF potenciómetro

```
/*
  ----------------------------------------
  Control ON/OFF con potenciómetro
  ----------------------------------------
*/
//-----------------------------------
//Función principal
//-----------------------------------
void setup() { // Se ejecuta cada vez que el Arduino se inicia
  Serial.begin(9600); //Inicia comunicación serial
  pinMode(13, OUTPUT); //Configurar el pin 13 como una salida
}

```

EJEMPLO 4: Control ON/OFF potenciómetro

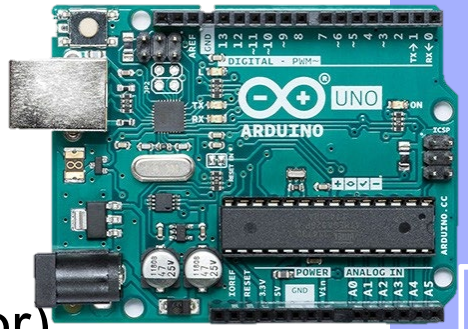
```
14  //---------------------------------
15  //Funcion ciclicla
16  //---------------------------------
17  void loop() {// Esta funcion se mantiene ejecutando
18
19    //Guardar en una variable el valor de la lectura análoga
20    int valor = analogRead(A0);
21    Serial.println(valor); //Imprime el valor por la consola
22
23    //Si el valor es mayor o igual a 500
24    if (valor >= 500)  {
25      digitalWrite(13, HIGH); //Enciende el LED en el pin 13
26    }
27    //Si el valor es menor a 500
28    else   {
29      digitalWrite(13, LOW); //Apaga el LED en el pin 13
30    }
31    delay(100); //Retardo de 100ms para ver los datos de la consola
32  }
33  //Fin programa
```

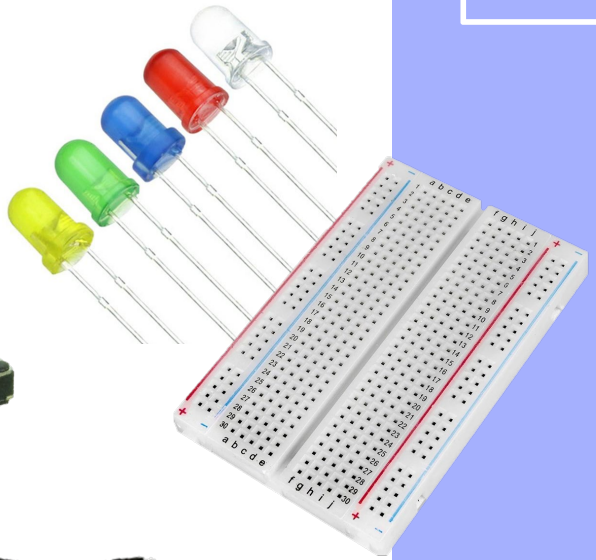
EJEMPLO 5: Lectura digital.
Contador de pulsos

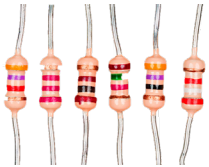


Materiales Ej. 5

- 1 Tarjeta Arduino
- 1 Led 5mm (cualquier color)
- 1 Resistencia (220-330)Ω
- 1Resistencia 10kΩ
- 3 Jumpers macho-macho (cables)
- 1Pulsador
- 1Protoboard







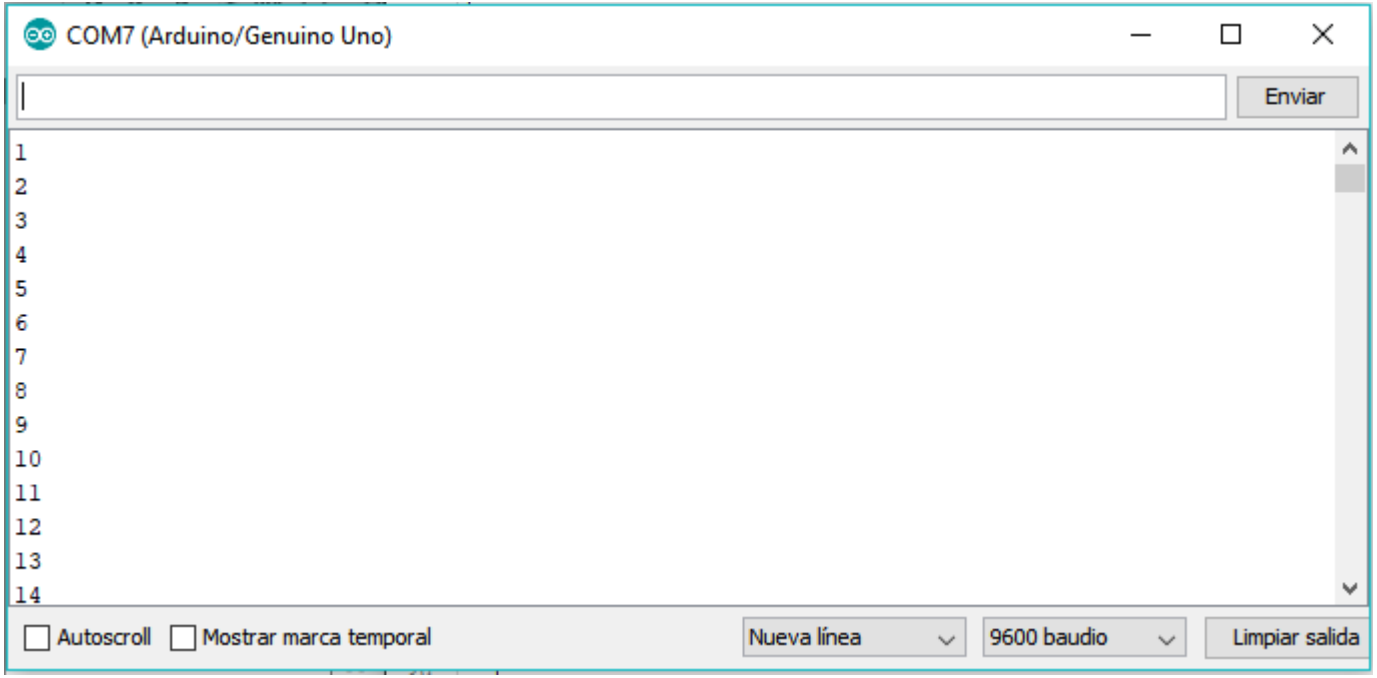
EJEMPLO 5: Contador de pulsos

```
/*
  ---------------------------------------------
  Contador de pulsos
  ---------------------------------------------
*/

//-------------------------------------------------
//Declara puertos de entradas y salidas y variables
//-------------------------------------------------
int conta = 0; //Variable para guardar el conteo de los pulsos
//-------------------------------------
//Función principal
//-------------------------------------
void setup() { // Se ejecuta cada vez que el Arduino se inicia
  Serial.begin(9600); //Inicia comunicación serial
  pinMode(2, INPUT); //Configura el pin 2 como una entrada, pulsador
  pinMode(13, OUTPUT); //Configura el pin 13 como una salida, LED
}

```

EJEMPLO 5: Contador de pulsos

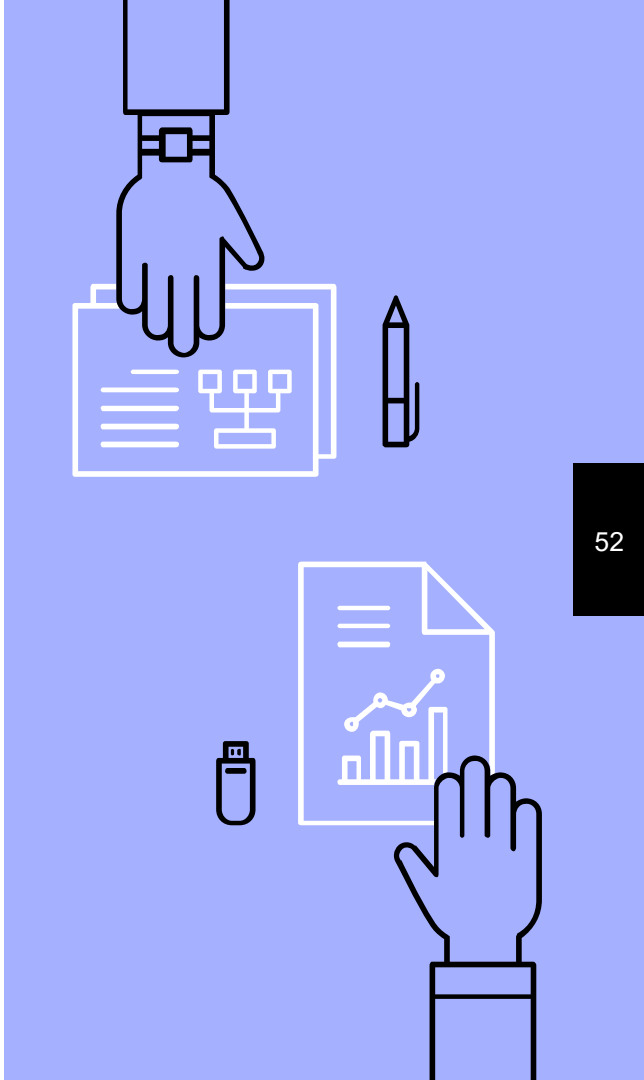
```
21  //-----------------------------------
22  //Función cíclicla
23  //-----------------------------------
24  void loop() { // Esta funcion se mantiene ejecutando
25    // Si el pulsador esta oprimido
26    if ( digitalRead(2) == HIGH )  {
27      // Si el pulsador no esta oprimido, flanco de bajada
28
29      if ( digitalRead(2) == LOW ) {
30        conta++; //Incrementa el contador
31        Serial.println(conta); //Imprime el valor por consola
32        delay (100); // Retardo
33      }
34    }
35
36    // Si el valor del contador es 5
37    if (conta == 5) {
38      digitalWrite(13, HIGH); //Enciende el LED
39    }
40    // Si el valor del contador es 8
41    if (conta == 8) {
42      digitalWrite(13, LOW); // Apaga el LED
43    }
44  }
45  //Fin programa
46
```

EJEMPLO 5: Contador de pulsos



LISTA DE MATERIALES MÓDULO 1

- 1 Tarjeta Arduino con cable USB
- 2 Leds (Cualquier color)
- 2 Resistencias (220-330)Ω
- 1 Resistencia 10kΩ
- 1 Potenciómetro (1-10k) Ω
- 1 Pulsador
- 5 Jumpers o más. (Cable protoboard)
- 1 Protoboard



¿Alguna pregunta?

Contacto al correo UNAL:

juacastropa@unal.edu.co

GRACIAS

LISTA DE ENLACES

1. *Link de materiales de este taller: https://github.com/JuliansCastro/Arduino/blob/master/README.md*
2. *Referencia del lenguaje: www.arduino.cc/reference/en/*
3. Blog sobre programación: https://programarfacil.com/blog/arduino-blog/curso-de-arduino/
4. Tienda de materiales: http://tdrobotica.co www.vistronica.com
5. Libro kit básico TdRobótica: https://issuu.com/tdrobotica/docs/libro_kit_basico/4
6. *Historia Arduino: https://arduinohistory.github.io/*
7. Icon made by becris from www.flaticon.com
8. *Imágenes libres: https://es.freeimages.com/ https://www.freepng.es/ https://thenounproject.com/* www.flaticon.com