

ARDUINO

* El proyecto Arduino:

Qué es y por qué usarlo

- [Arduino](#) es un sistema electrónico de prototipado abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Con conceptos básicos de electricidad y electrónica, curiosos artistas, diseñadores..., pueden crear objetos o entornos interactivos digitales.
- OBJETIVOS: aprendizaje y capacidad para resolución de problemas.
- [Físicamente](#) y para fines prácticos arduino es:
 - una placa que contiene un microprocesador
 - pines de entrada y salida para la transmisión y recepción de datos digitales o analógicos.
 - una conexión USB que permite cargarle programas y comunicarse con el PC.

* Dispone de un [IDE](#) (Entorno de desarrollo) cómodo y flexible.

¿Qué podemos hacer con Arduino?

- Arduino se puede utilizar para desarrollar **objetos interactivos** autónomos o puede ser conectado a software de un PC, por ejemplo Processing, y establecer órdenes y respuestas con el exterior.

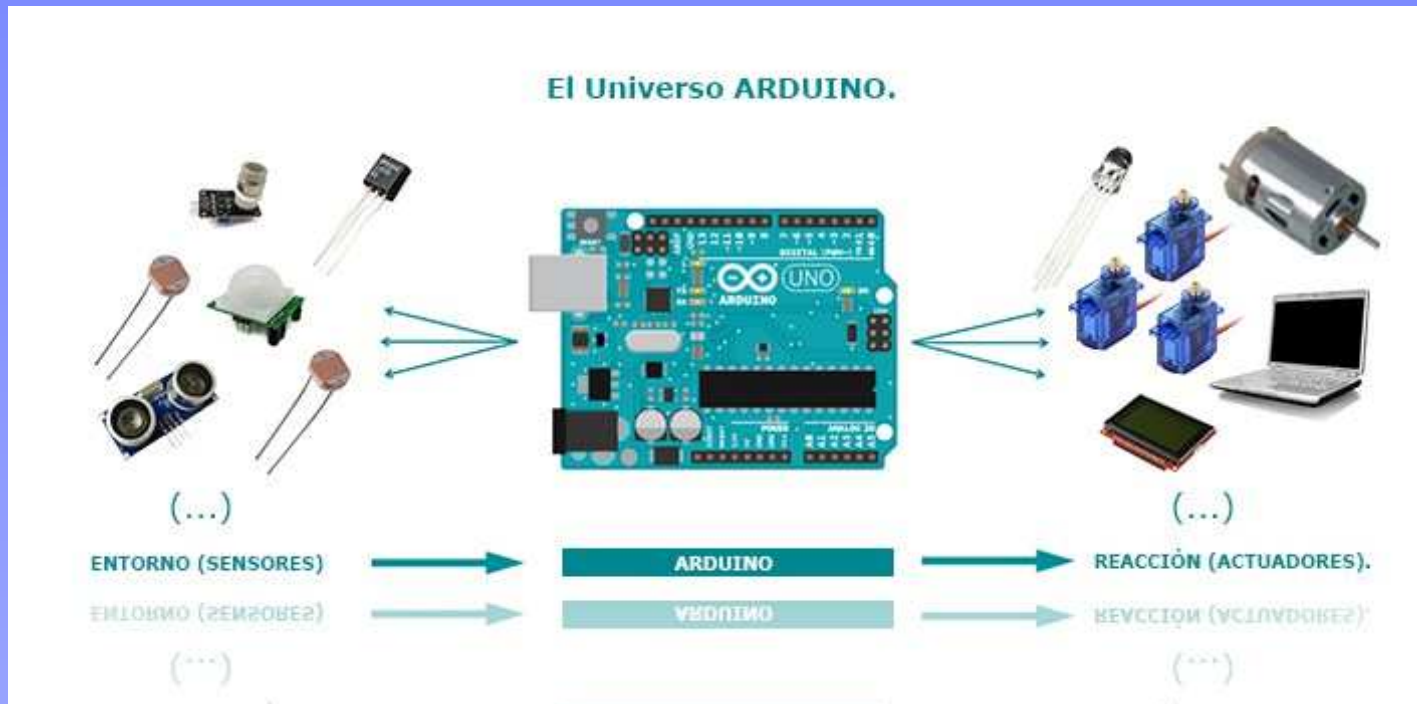
Cosas que podemos hacer:

- Controlar sistemas a partir de entradas y salidas

Controlar luces, motores, timbres y otros actuadores a partir de sensores como pulsadores, LDR, sensores de distancia, de presión...

- Construir robots que funcionen de forma automática
- Conectar programas con la realidad mediante sensores

COMENZAMOS...

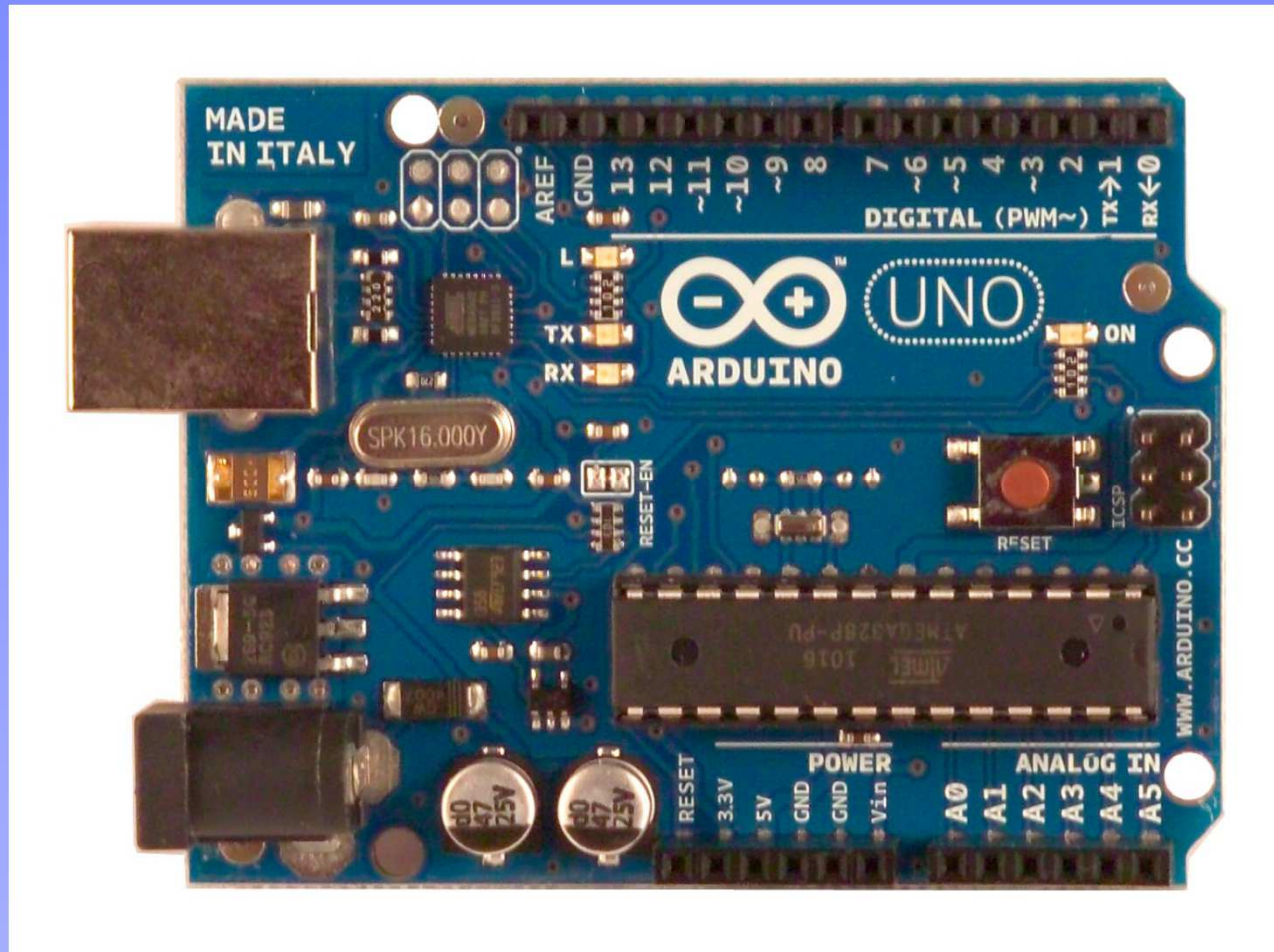


- <http://www.arduino.cc> o en su versión en español: <http://www.arduino.cc/es/>

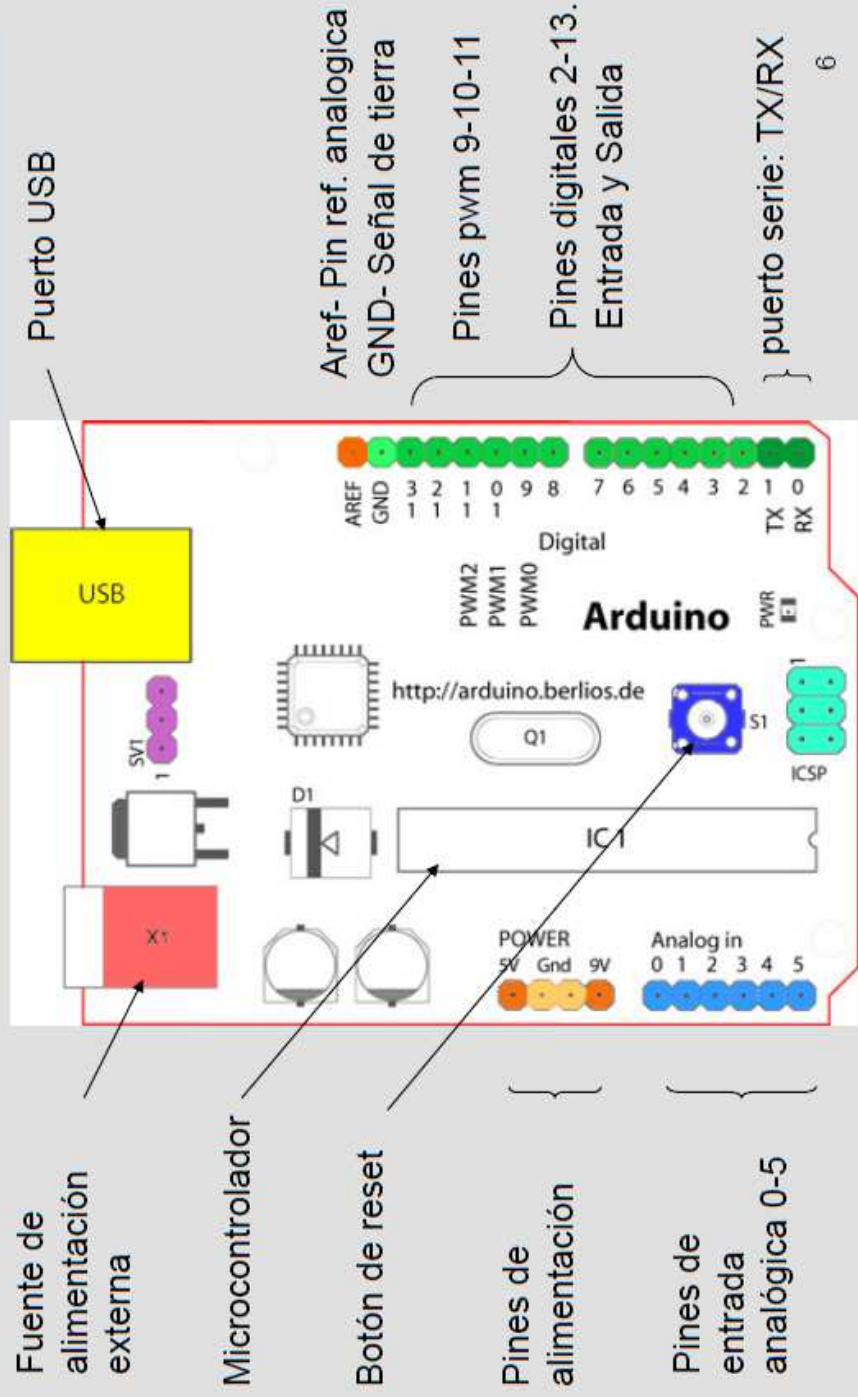
<https://www.arduino.cc/en/Reference/HomePage>

<https://processing.org/tutorials/electronics/>

PLACA ARDUINO

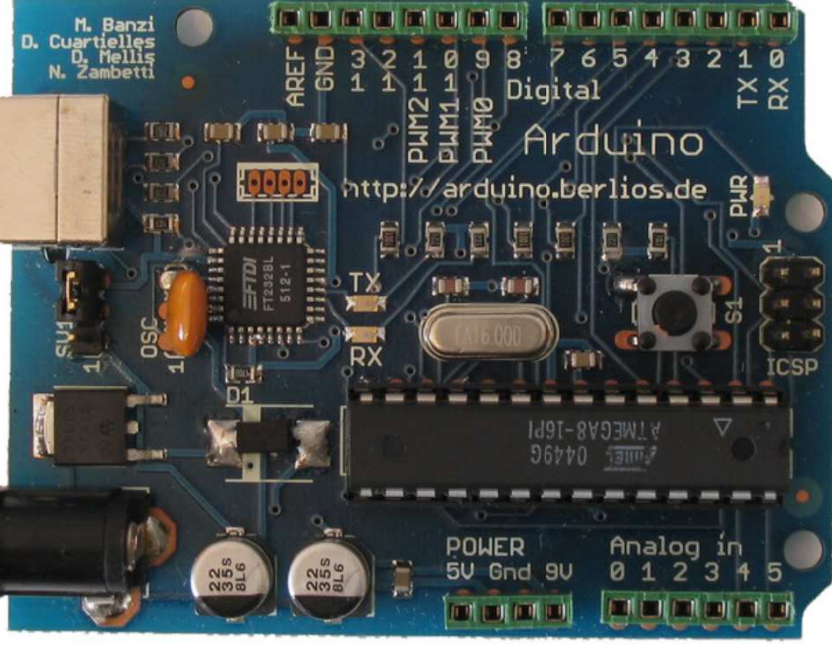


Descripción de la placa



Entrada Fuente Alimentacion
externa

Puerto de conexion
USB



Señales de Alimentacion
5v, 0v y 9v

Pines Analogicos
de entrada

Pines Digitales
de Entrada y Salida 2-13
Pines pwm
9-10-11

Pines de comunicacion
Tx-Rx

Foto de familia: Las variantes de Arduino.

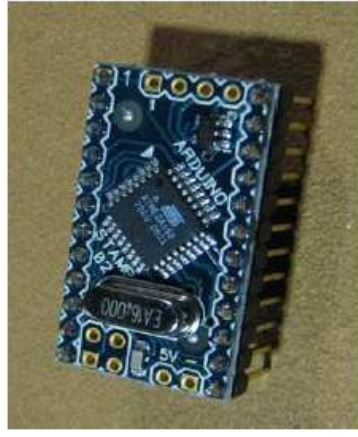


Ilustración 5: Arduino Mini



Ilustración 4: Arduino Bluetooth

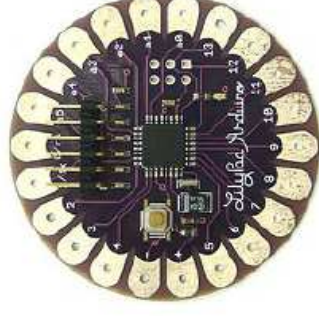


Ilustración 3: Arduino Lilypad

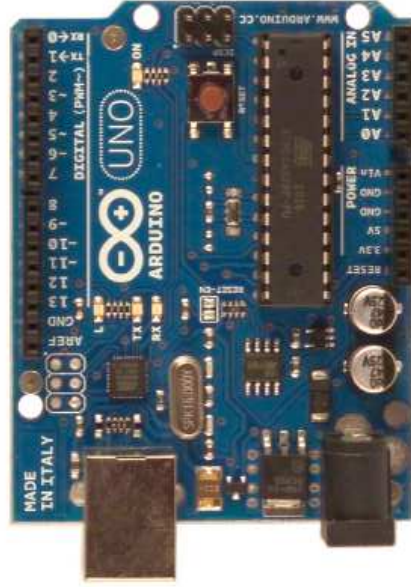


Ilustración 11: Arduino UNO



Ilustración 12: Arduino UNO SMD

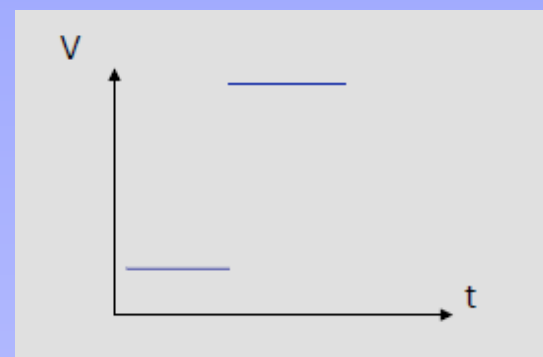
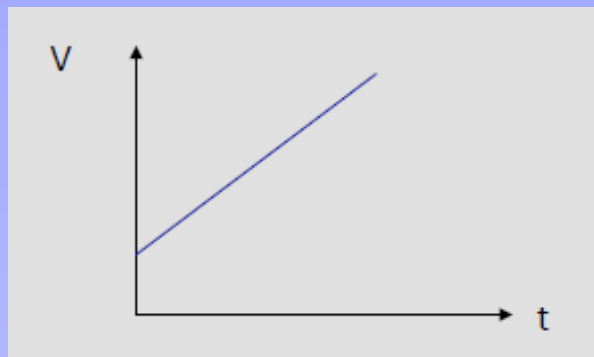
Señales: Analógicas y Digitales

- Transmisión de información y señal eléctrica.

En el formato analógico la transformación de datos está formada por impulsos eléctricos con amplitud de valores continuos.

En el caso de señales digitales los valores se ordenan en torno a un código por valor de 0 y 1, donde cada bit es una representación de dos amplitudes distintas. Posteriormente es reconstruido a través del software oportuno.

Característica común a los ordenadores que utilizan: '1' y '0'.



Arduino representa el '1' con una señal de 5 voltios y el '0' con una señal de 0 voltios.

En programación llamamos una señal de 5 voltios, HIGH (significa ALTA)
0 voltios LOW (BAJA).

*** Cuando Arduino escribe en un Pin de salida **HIGH**, pondrá 5 voltios en ese Pin, se pondrá en **ON** (se encenderá).

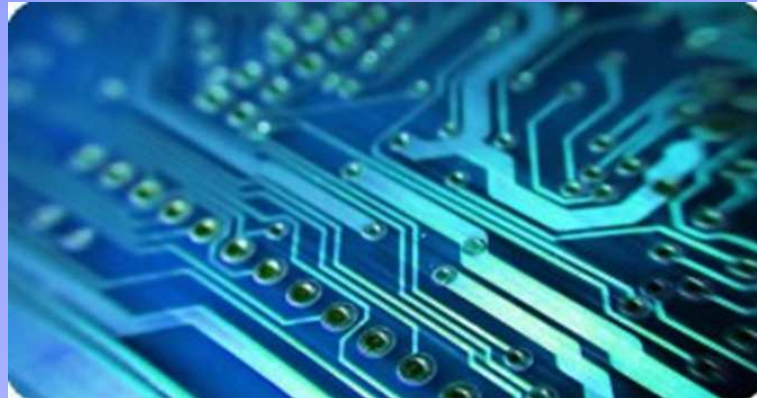
*** Cuando escribimos **LOW** en el Pin, Arduino pondrá el Pin a 0 voltios y estará en **OFF** (lo apagará).

Señales analógicas:

En lugar de 2 estados opuestos, tienen niveles continuos.

En Arduino, puedes obtener los valores analógicos de los Pins analógicos.

Sobre la placa puedes ver un grupo de Pins marcados como **analog in**. Cuando se les aplica tensión, ellos reportan valores de 0 a 1023;



Entradas



Dan información al sistema, son los sensores:

- Pulsador
- Potenciómetro
- LDR

Microcontrolador

Procesa la información que recibe de las Entradas y transmite una actuación a las Salidas

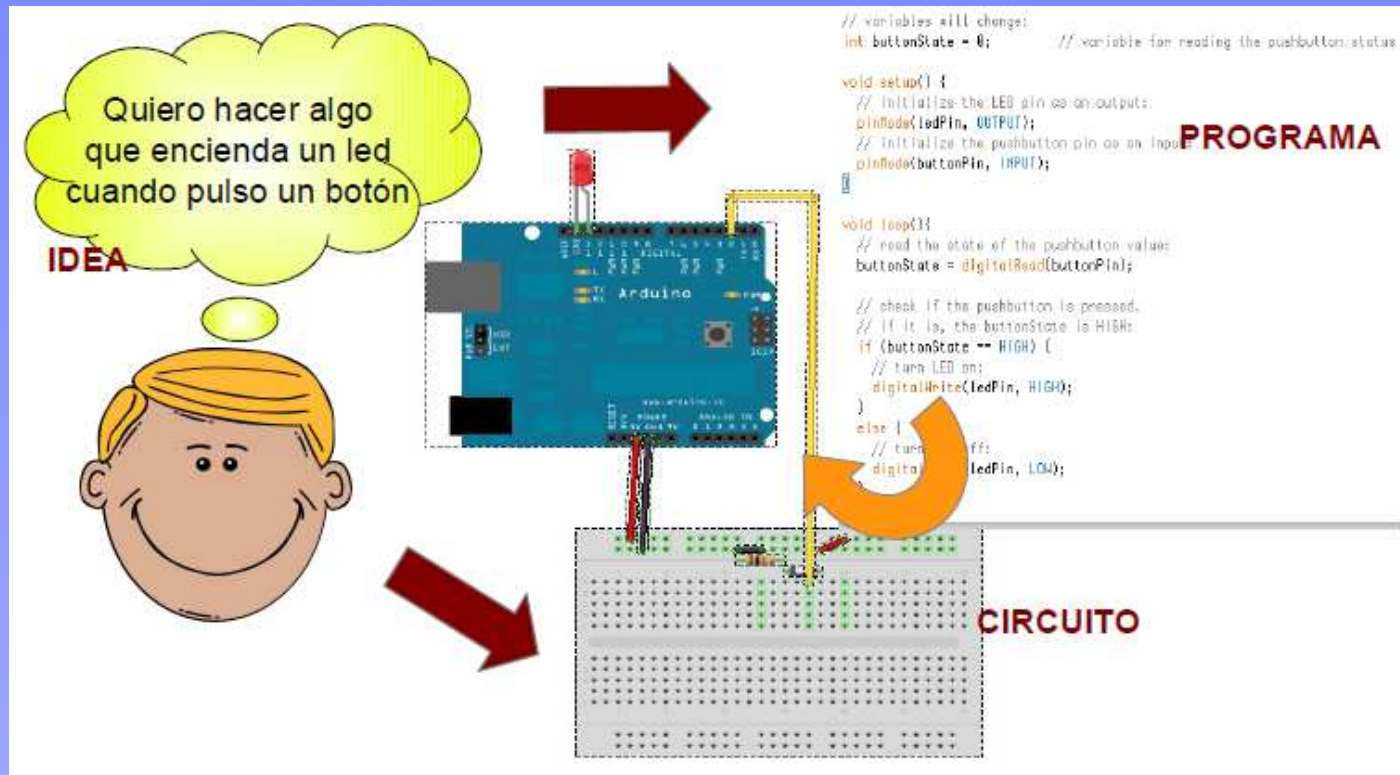
Salidas



Producen una actuación:

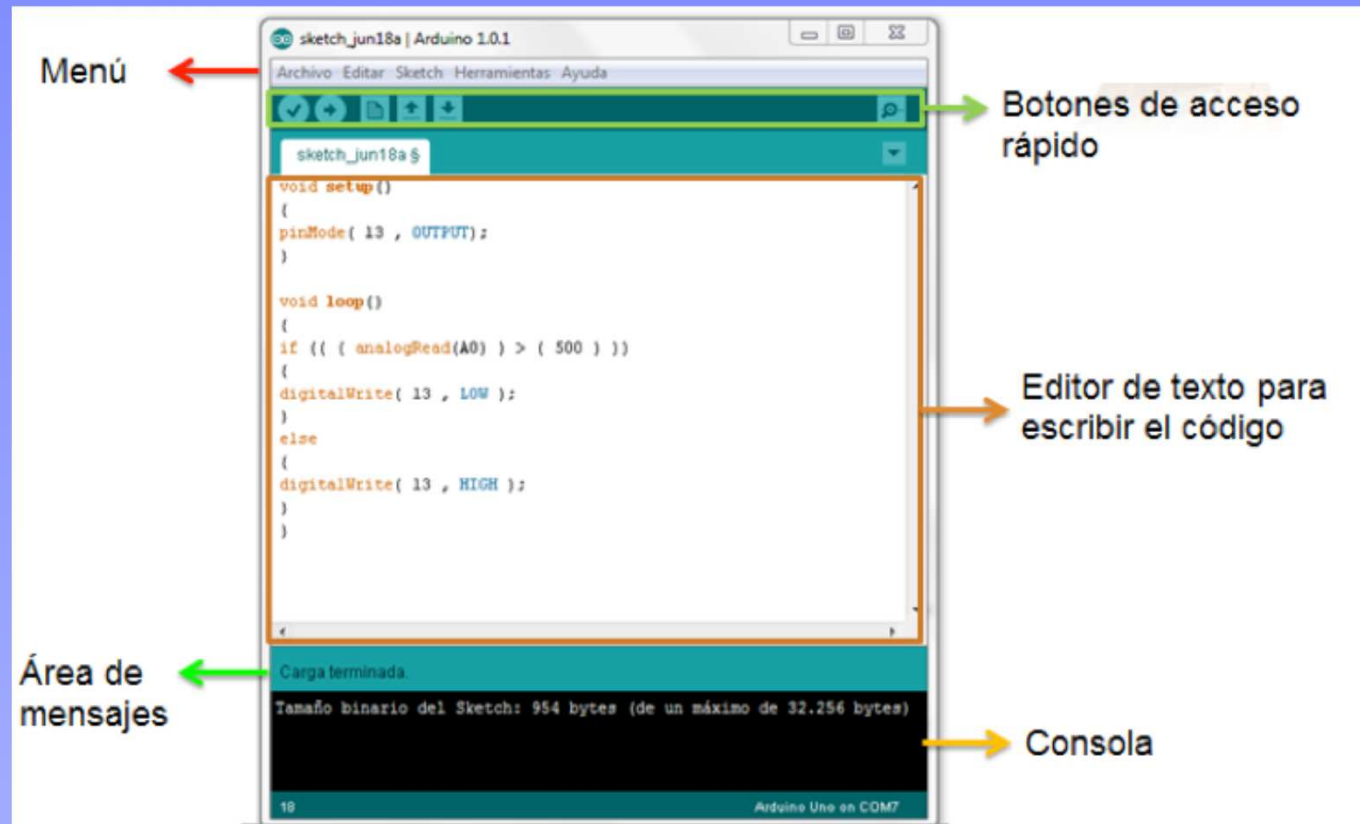
- Led
- Zumbador
- Motor

ENTORNO IDE ARDUINO

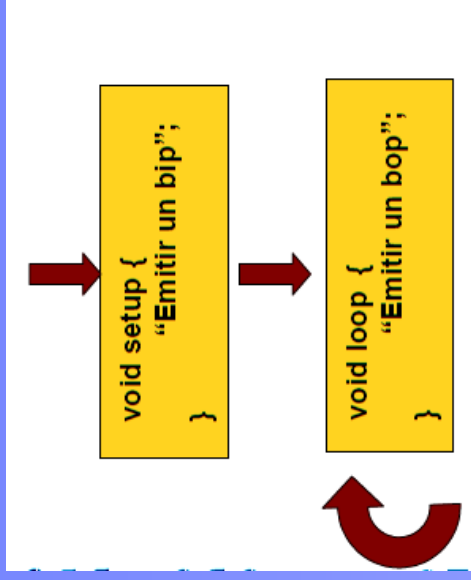


Cuando trabajamos con arduino, creamos un circuito. Plasmamos una idea en un programa, esta se graba en la placa y a través de él lo verificamos y ejecutamos.

- Barra de menús
- Botones de acceso
- Barra de mensajes
- Editor
- Consola



- ✓ verificar si tu programa va a funcionar
- cargar el programa a la placa de Arduino
- 📊 crear un programa nuevo
- ↑ abrir un programa
- ↓ guardar el programa en el disco duro del ordenador



Estructura del programa.

Declaración de variables: en esta parte declaramos todas las variables que vamos a usar en el programa. Nos permite asociar nombres a números lo que nos será de mucha utilidad a la hora de modificar los programas.

Void setup{ } es la función de **configuración** de los pines de Arduino y sólo se ejecuta una vez, mientras que loop() se ejecuta una y otra vez hasta que apaguemos el sistema, o se gasten las baterías

Bucle infinito: Void loop{ } es la parte del programa que se ejecuta de forma cíclica.



sketch_prueba_LED_ARDUINO_SOLO

```
void setup(){
  //configurar pin 13
  pinMode(6, OUTPUT);
}

void loop(){
  //encender, esperar 2 segundos
  digitalWrite(6, HIGH);
  delay(1000);

  //apagar, esperar 3 segundos
  digitalWrite(6, LOW);
  delay(1000);
}
```

Elementos del programa

Temporizadores

- **delay(ms tiempo)**, sirve para parar los procesos de la placa durante un cierto *tiempo* en milisegundos

Fragmento de código

```
delay(1000);
```

Pines digitales

- **pinMode(pin, modo)** sirve para declarar un pin como entrada (INPUT) o como salida (OUTPUT)

Configuración de los pines dentro del bloque setup (sólo una vez).

Fragmento de código

```
pinMode (13,OUTPUT); //establece el pin 13 como salida  
pinMode (15,INPUT); //establece el pin 15 como entrada
```


- **digitalWrite(pin, valor)**, sirve para escribir un *valor* al *pin*, el *valor* podrá ser 1 lógico (HIGH) o 0 lógico (LOW)

Los dos estados irán asociados a un nivel de 5 voltios (alto), o 0 voltios (bajo).

Fragmento de código

```
entrada = digitalRead(10); //asocia la variable "entrada" al pin digital 10
digitalWrite (10, HIGH); //establece el pin 10 con estado HIGH
```



ANALOGWRITE Y ANALOGREAD

A diferencia de los digitales, los pines analógicos pueden tomar cualquier valor entre 0 y 5 voltios. Para escribir un valor analógico se usa una PWM (siglas en inglés Modulación por Ancho de Pulso).

Fragmento de código

```
entrada = analogRead (1); //Asocia a la variable "entrada" el valor del pin 1
analogWrite (11, 128); //Direcciona al pin 11(es de tipo PWM) el valor de
tensión correspondiente a 128 (cercano a los 2,5 voltios)
```

FUNCIONES SERIAL

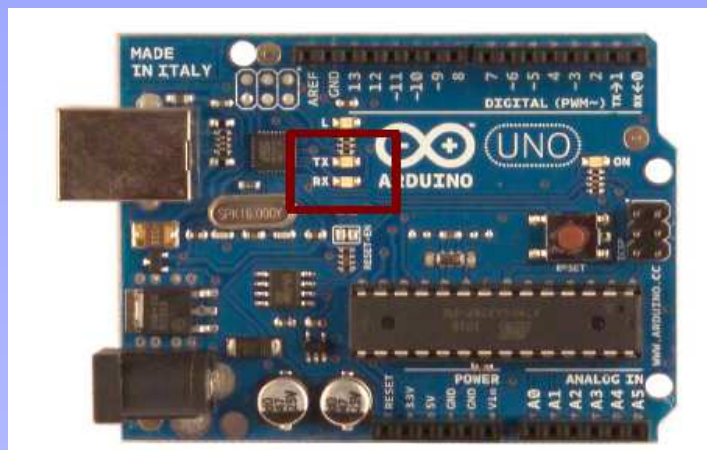
Las placas de Arduino poseen, en principio, un puerto serie para la comunicación con la computadora u otros dispositivos. Tal comunicación se produce en los pines TX y RX.

- La instrucción **Serial.begin()** inicializa el puerto serie y establece la velocidad de comunicación (especificada en baudios).

Fragmento de código

```
Serial.begin(9600); //abre el puerto serie y establece la velocidad de comunicación en 9600bps
```

Inicializa la comunicación interna del microcontrolador, indicando qué velocidad vamos a transmitir. Cuanto más grande sea el número, más bits por segundo pasarán por el cable.



Constantes

El lenguaje Arduino tiene unos cuantos valores predefinidos que se llaman constantes. Se usan para hacer los programas más legibles. Las constantes se clasifican en grupos.

true/false

Estas son constantes Booleanas que definen niveles lógicos. FALSE se define como 0 (cero) mientras TRUE es 1 o un valor distinto de 0.

```
if (b == TRUE)
{
    doSomething;
}
```

high/low

Estas constantes definen los niveles de pin como HIGH o LOW y se usan cuando se leen o se escriben los pines digitales. HIGH esta definido como el nivel 1 lógico, ON ó 5 V, mientras que LOW es el nivel lógico 0, OFF ó 0 V.

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

input/output

Constantes usadas con la función *pinMode()* para definir el modo de un pin digital como INPUT u OUTPUT.

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

El programa:

```
void setup() {  
  // Ponemos el pin 12 en modo salida.  
  pinMode(12, OUTPUT);  
  
}  
  
void loop() {  
  // Lanzamos 5 Voltios por el pin 12.  
  digitalWrite(12, HIGH);  
  // Esperamos 1 segundo.  
  delay(1000);  
  // Dejamos el puerto 12 a 0 V.  
  digitalWrite(12, LOW);  
  // Esperamos 1 segundo.  
  delay(1000);  
}
```

¿Qué queremos decir con todo esto?

→ SIMILITUDES ←

bloques
funciones

...

SINTAXIS DEL PROGRAMA:

Arduino.list(): devuelve una lista con los dispositivos serie (puertos serie) disponibles. Si su tarjeta Arduino está conectada a la computadora cuando usted llama a esta función, su dispositivo estará en la lista.

Arduino(parent, name, rate): crea un “objeto” Arduino (objeto a nivel de elemento de programación). *parent* debe aparecer sin comillas; *name* es el nombre del dispositivo serie (es decir, uno de los nombres devueltos por `Arduino.list()`); *rate* es la velocidad de la conexión (57600 para la versión actual del de firmware).

pinMode(pin, mode): *pin* configura un pin digital como entrada (input) o como salida (output) *mode* (`Arduino.INPUT` o `Arduino.OUTPUT`).

digitalRead(pin): devuelve el valor leído de una de las entradas digitales, `Arduino.LOW` o bien `Arduino.HIGH` (el pin debe estar configurado como entrada).

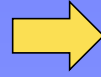
digitalWrite(pin, value): escribe `Arduino.LOW` o `Arduino.HIGH` en un pin digital.

analogRead(pin): devuelve el valor de una entrada analógica leída (de 0 a 1023).

analogWrite(pin, value): escribe un valor analógico (señal tipo PWM) en un pin digital que soporta salida analógica (pines 3, 5, 6, 9, 10, y 11 para ATMEGA 168); valores debes estar comprendidos entre 0 (equivalente a off) y 255 (equivalente a on).

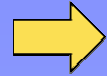
...

llaves {}



```
void loop()
{
  statements;
}
```

punto y coma ;



```
int x = 13;
```

bloques de comentarios /*...*/

comentarios de línea //

Variables

Bucles

Condicionales

. DE COMPARACIÓN

Operador	Descripción
==	x == y, ¿es x igual a y?
!=	x != y, ¿es x distinto de y?
<	x < y, ¿es x menor que y?
>	x > y, ¿es x mayor que y?
<=	x <= y, ¿es x menor o igual que y?
>=	x >= y, ¿es x mayor o igual que y?

. OPERADORES ARITMÉTICOS

Operador	Descripción
=	a = b, asignación
+	a + b, adición o suma
-	a - b, sustracción o resta
*	a * b, multiplicación
/	a / b, división
%	a % b, módulo (resto de la división)

OPERADORES COMPUESTOS

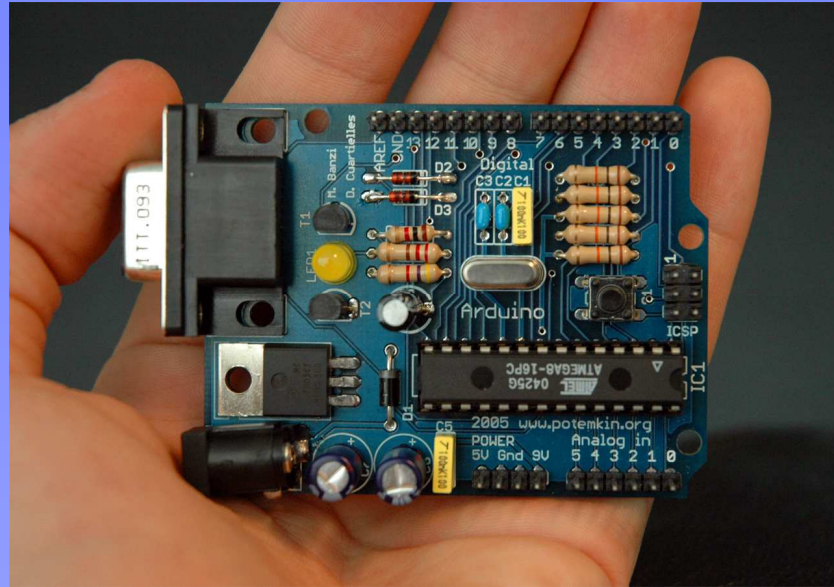
Operador	Descripción
++	a++, incremento (suma 1)
--	a--, decremento (resta 1)
+=	a += b, adición compuesta
-=	a -= b, sustracción compuesta

OPERADORES BOOLEANOS

Operador	Descripción
!	!a, negación
&&	a && b, "y" lógico
	a b, "o" lógico

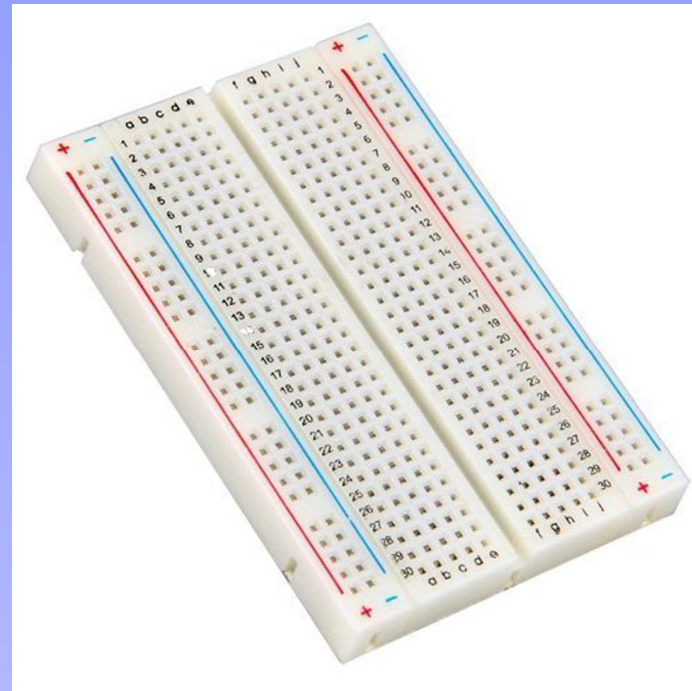
MATERIALES

Placa arduino

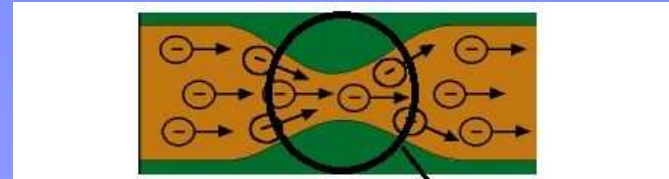


Placa de prototipos
para el armado y prototipado de
circuitos electrónicos

[https://www.youtube.com/
watch?v=ulsv7uxDZoc](https://www.youtube.com/watch?v=ulsv7uxDZoc)



Resistencias:



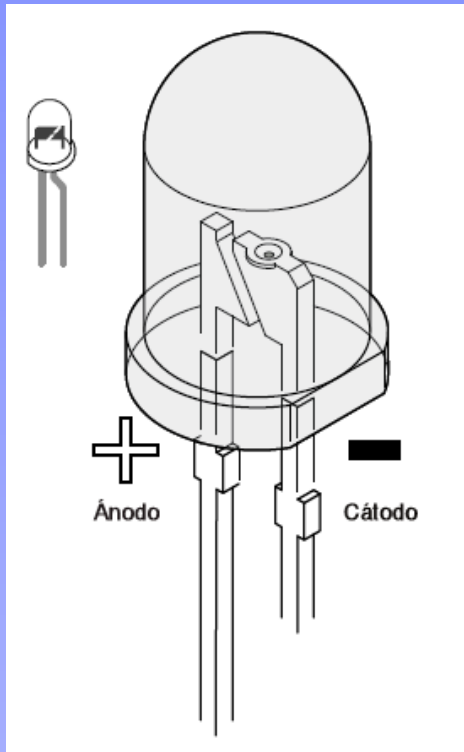
La **Resistencia Eléctrica** es la oposición o dificultad al paso de la corriente eléctrica. Se mide en Ohmios (Ω) y se representa con la letra R.

Normalmente para un led se utiliza resistencia de 220 ohmios.

<http://www.digikey.es/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band>

LEDS:

Es un emisor de Luz.



Los LEDs son construidos para mostrar esto (LEDs are constructed to show this). El Pin largo, se denomina ánodo, y es el positivo, cuando lo (wiring/ cableamos) conectamos al circuito éste debe estar próximo al extremo positivo del alimentador.

El Pin corto, o cátodo, debe estar lo más cerca posible a la tierra (0 voltios).

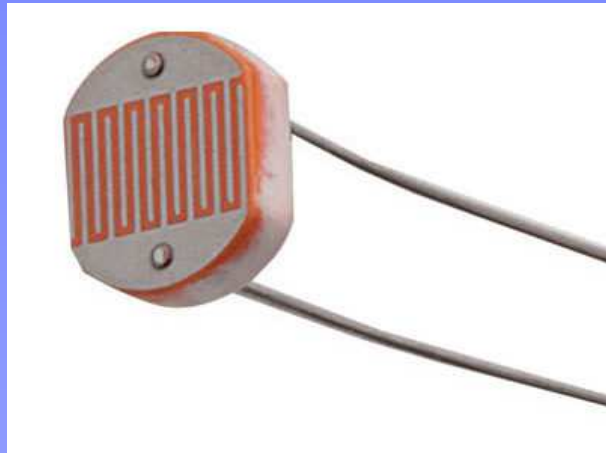
Transistores:



Un transistor es un dispositivo que regula el flujo de corriente o de tensión actuando como un interruptor o amplificador para señales electrónicas.

También se llama **Transistor Bipolar** o **Transistor Electrónico**.

Fotoresistencia o resistencia LDR



Es un componente electrónico cuya resistencia disminuye o aumenta con el la intensidad de luz incidente.

El valor varía en función de la luz a la que se exponga.

Potenciometros:

Resistencias variables que actúa como medidor de tensión o voltaje.



Pulsador:

Dispositivos para circuitos electrónicos. Permiten el flujo de corriente cuando son activados. Unen los extremos de sus patas al ser presionados para permitir el paso de corriente eléctrica.



Relés:

Son interruptores que podemos activar mediante señal eléctrica. Nos van a permitir trabajar con mayores voltajes y por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.



https://www.arduineando.com/tutoriales_arduino/

DESCARGA EL ENTORNO ARDUINO

<https://www.arduino.cc>

Download the Arduino IDE



ARDUINO 1.8.2

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

Windows ZIP file for non admin install

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

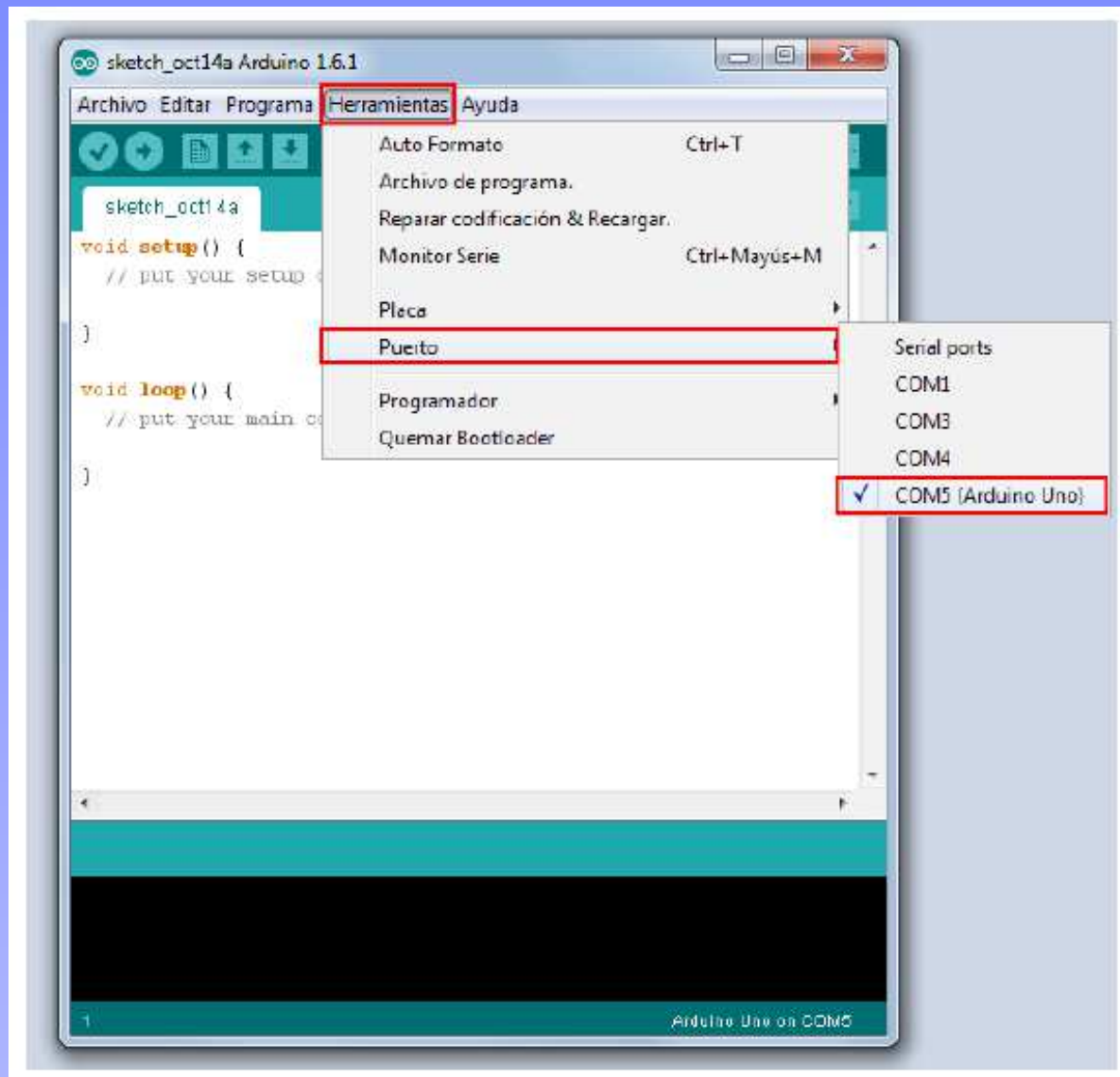
[Checksums \(sha512\)](#)

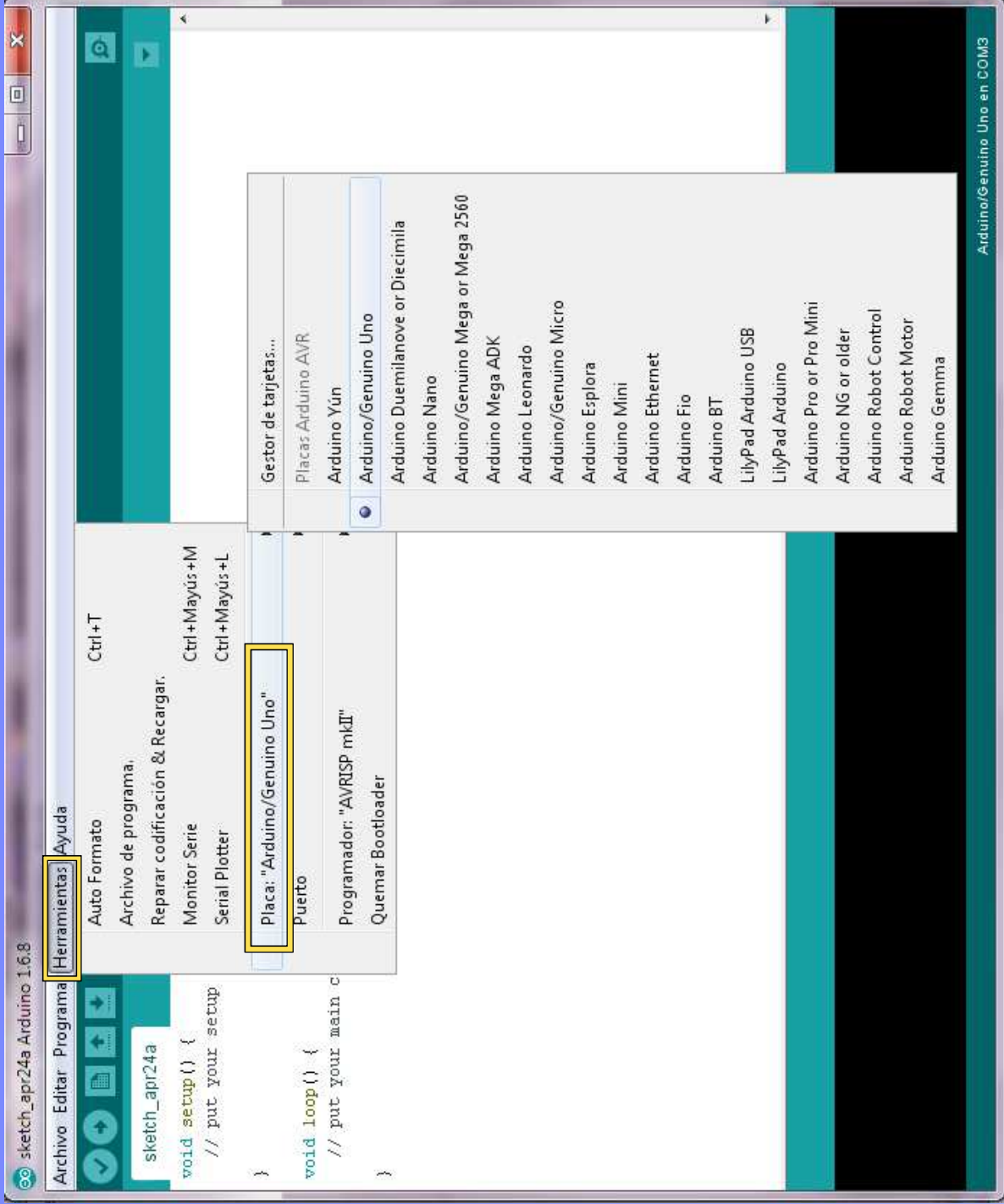
Instalación

Conecta Arduino al PC con USB

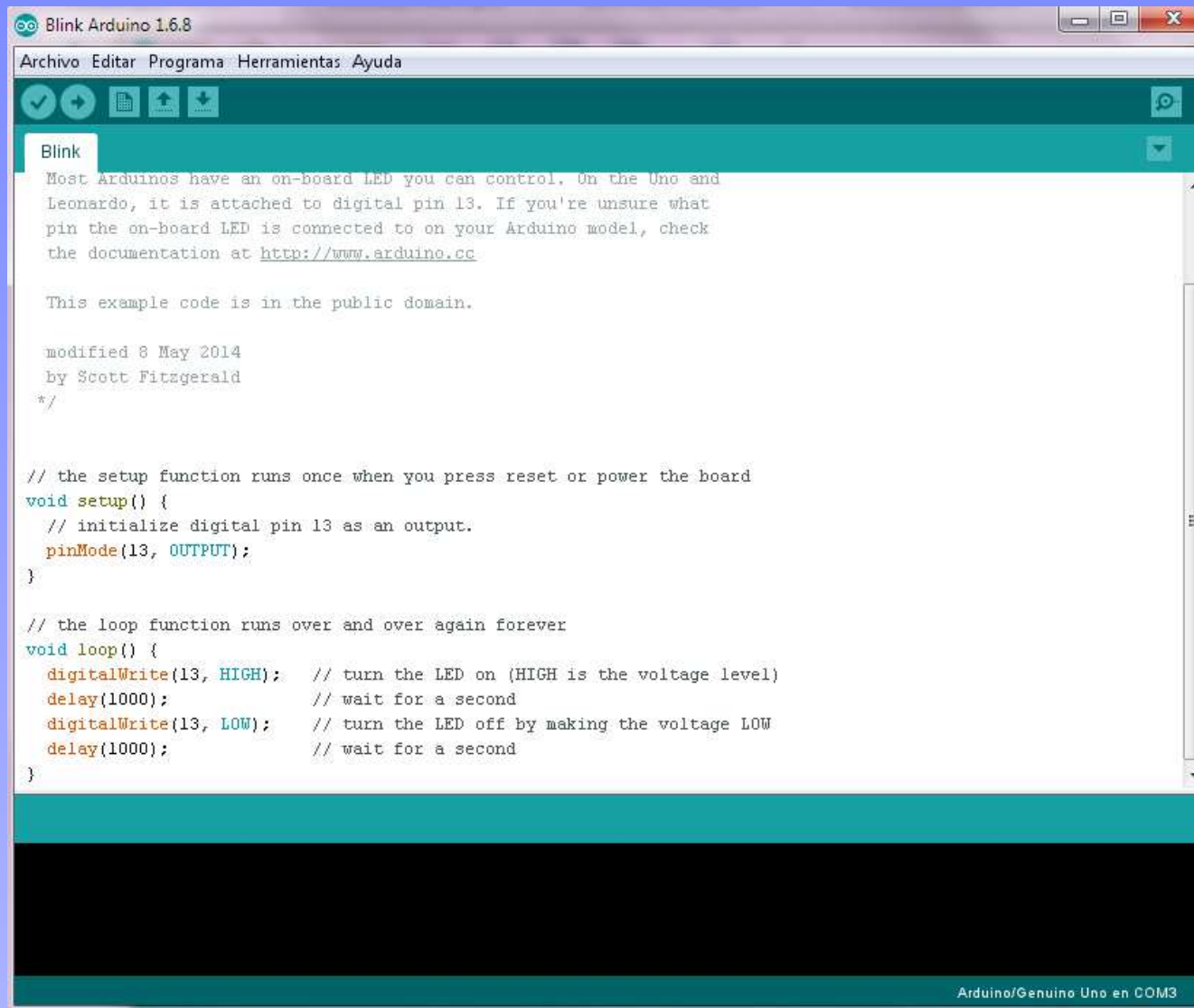
Ejecuta y abre la IDE de ARDUINO

Verificar el puerto de comunicación y la placa :





EJEMPLO DE LED INTEGRADO EN LA PLACA: el pin 13 (L)

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink Arduino 1.6.8". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for saving, undo, redo, and uploading. The main text area contains the following text and code:

Blink

Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at <http://www.arduino.cc>

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

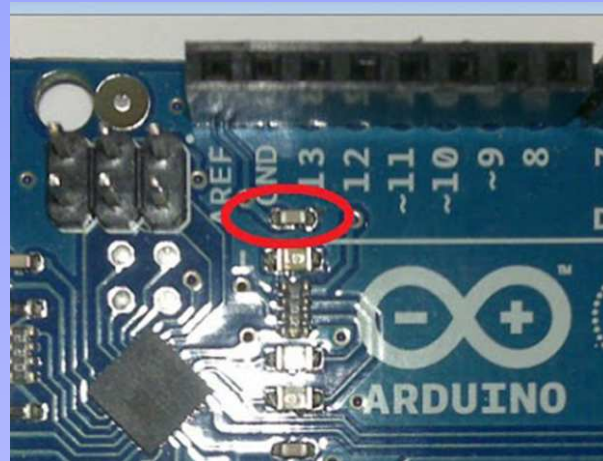
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second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

At the bottom right of the IDE, it says "Arduino/Genuino Uno en COM3".

Comprueba si la placa tiene actividad o no

El programa.

```
void setup() {  
  // Ponemos el pin 13 en modo salida.  
  pinMode(13, OUTPUT);  
  
}  
  
void loop() {  
  // Lanzamos 5 Voltios por el pin 13.  
  digitalWrite(13, HIGH);  
  // Esperamos 1 segundo.  
  delay(1000);  
  // Dejamos el puerto 13 a 0 V.  
  digitalWrite(13, LOW);  
  // Esperamos 1 segundo.  
  delay(1000);  
}
```



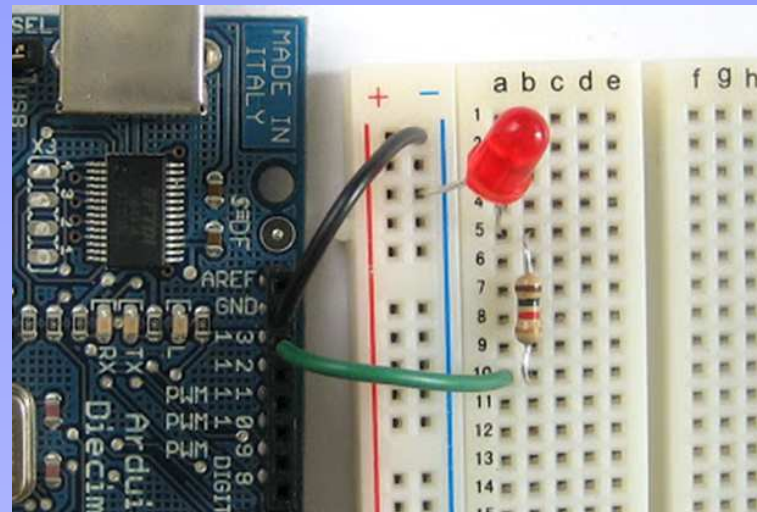
Verificar/compilar el sketch
Cargar el programa a la placa

MATERIALES:



Diodo led.
Cable USB tipo A-B
Resistencia 220ohms
Placa Arduino
Cables

Software



¿practicamos con tres LEDS?...