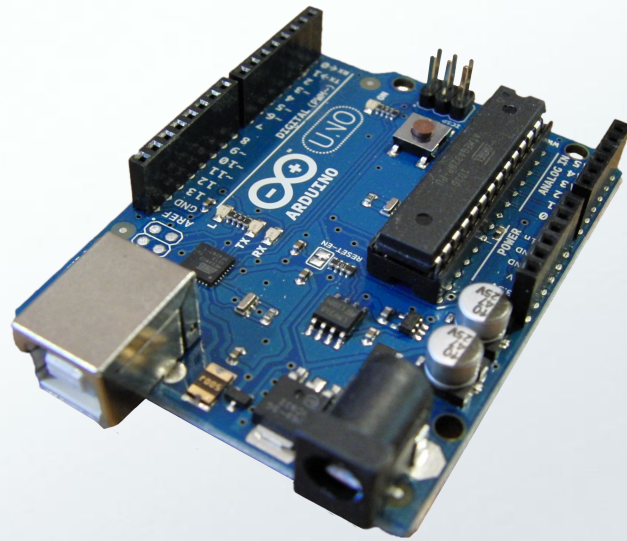


ARDUINO



¿QUÉ ES ARDUINO?

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing.

Es un dispositivo que conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital.





SUS CREADORES

David Cuartielles

Massimo Banzi

Alternativa económica a placas usadas por estudiantes





ARDUINO TOMA INFORMACIÓN DEL ENTORNO FÍSICO

Mediante sus puertos de entrada y usando sensores como luces, motores, pantallas...



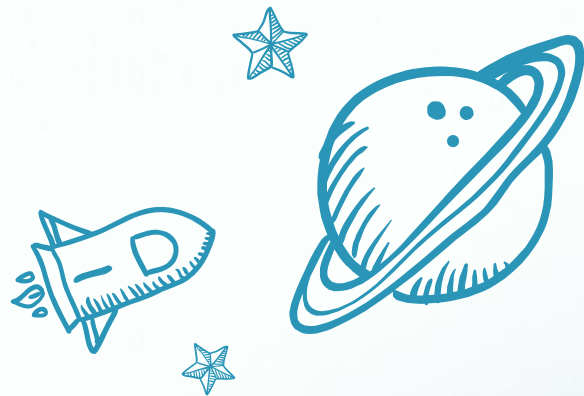


La placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing)



Las placas pueden ser hechas a mano o comprarse montadas de fábrica; el software puede ser descargado de forma gratuita.





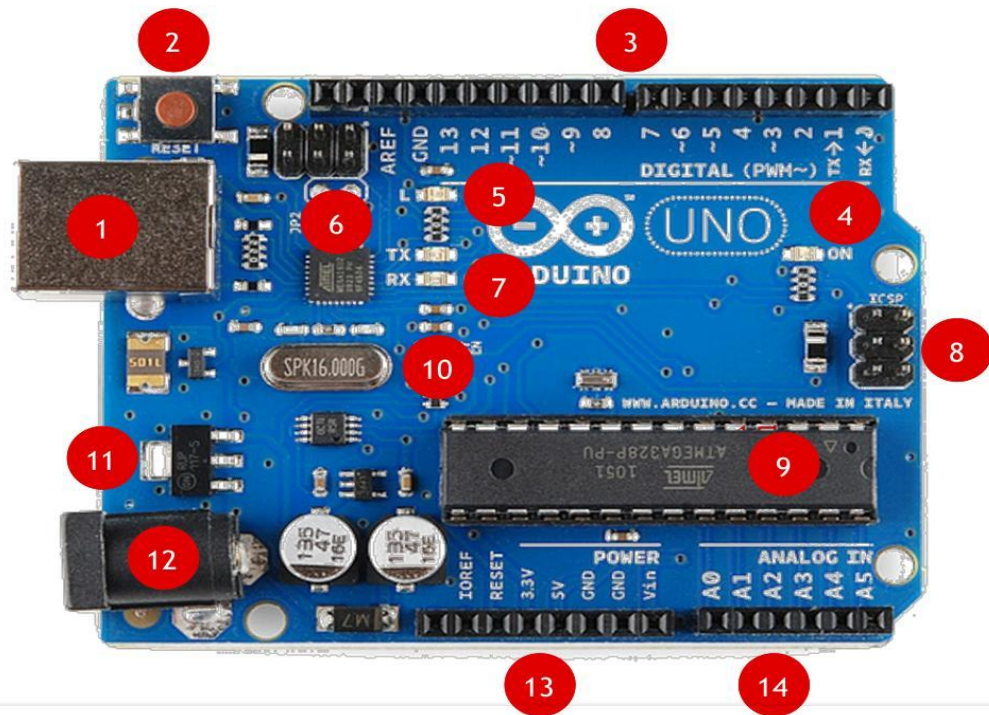
PERO, ¿COMO ES
UN ARDUINO?

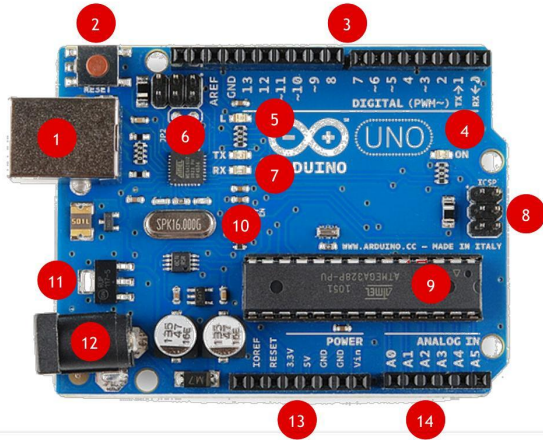




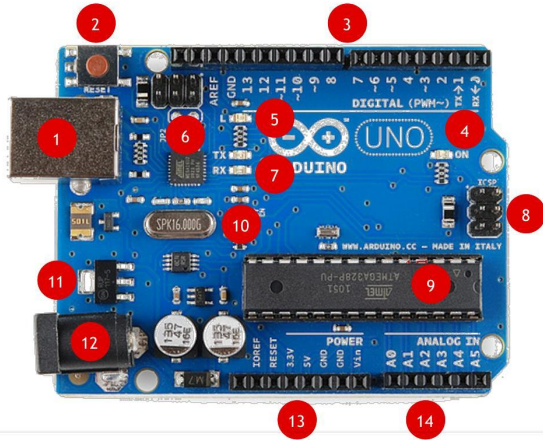
Es una placa electrónica basada en el **ATmega328P**. Cuenta con 14 puertos digitales de entrada/salida, 6 entradas analógicas, un resonador de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio.







1. Conector USB
2. Botón Reset
3. Pines E/S digitales y PWM
4. Led verde Placa Encendida
5. Led naranja conectada Pin 13
6. ATmega 16U2 encargado comunicación con PC
7. Led TX (Transmisor) y RX (Receptor) de la comunicación serial



8. Puerto ICSP para programación serial

9. Microcontrolador ATmega 328

10. Cristal de cuarzo de 16Mhz

11. Regulador de voltaje

12. Conector voltaje 9 a 12v

13. Pines de voltaje y tierra

14. Entradas análogas

INSTALACIÓN DE ARDUINO

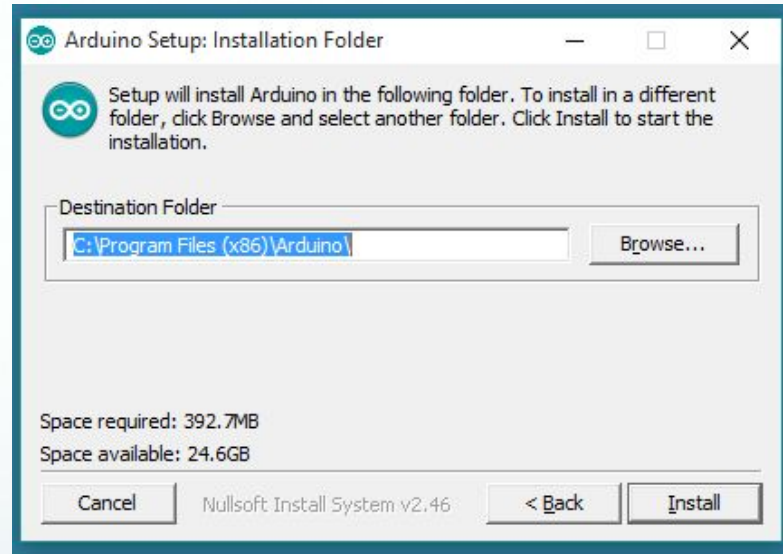
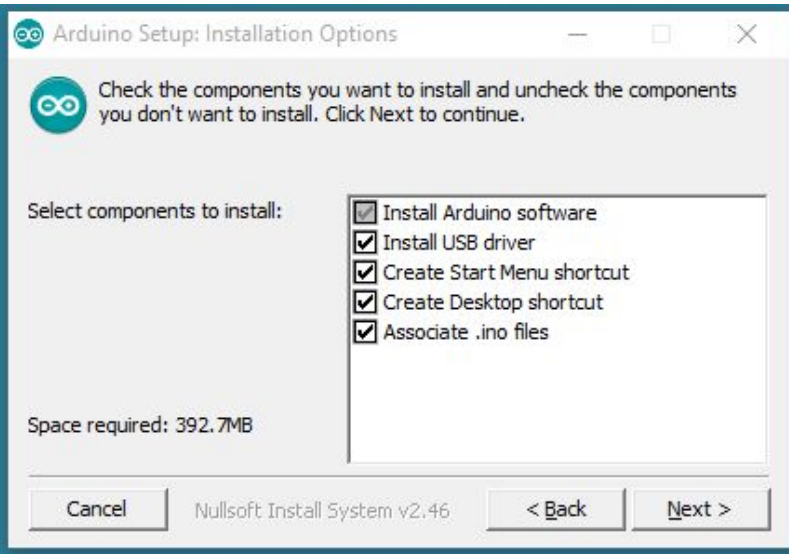
- Windows
- Mac OS X
- Linux

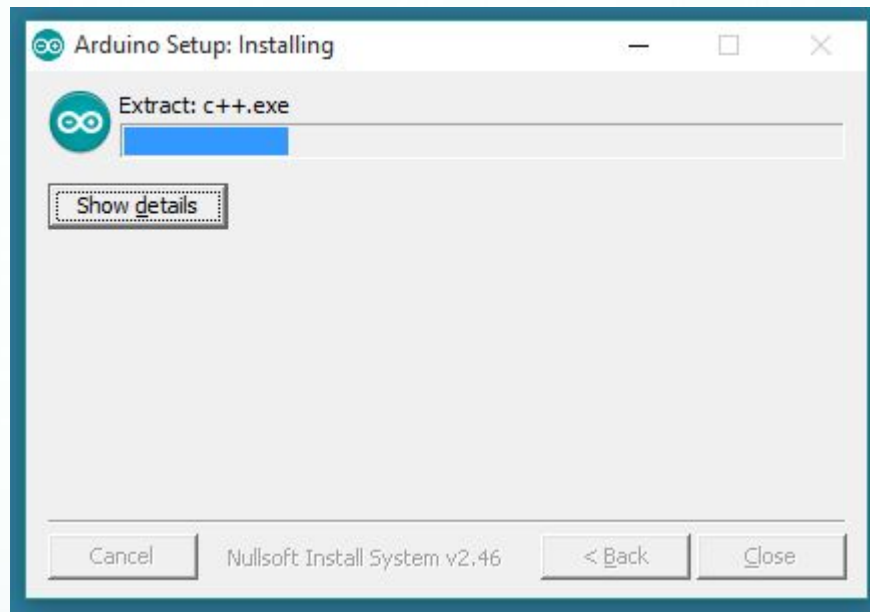


WINDOWS (VISTA, 7 , 8 , 10)

1. Descargar el software de :
www.arduino.cc/en/Main/Software
2. Doble clic en el instalador de nuestra versión de Windows. Aceptamos los términos y condiciones pulsando sobre "I Agree".







MAC OS X (VERSIÓN 10.8 O SUPERIOR)

1. “Preferencias del sistema” y panel de “Seguridad&Privacidad”. En la pestaña “General”, pulsamos “Permitir descargar aplicaciones desde” y clic en “Desde cualquier lugar”



2. Descargar el software de :
www.arduino.cc/en/Main/Software para MAC
OS X.

3. Descomprimos y copiamos la aplicación
Arduino a la carpeta Aplicaciones o donde
deseamos instalarla.

4. Conectamos la placa Arduino.



5. Dependiendo del tipo de versión de sistema operativo, podría aparecer una ventana preguntando si desea abrir las “Preferencias del sistema”. Hacemos clic en “Preferencias de Red” y después en “Aplicar”.

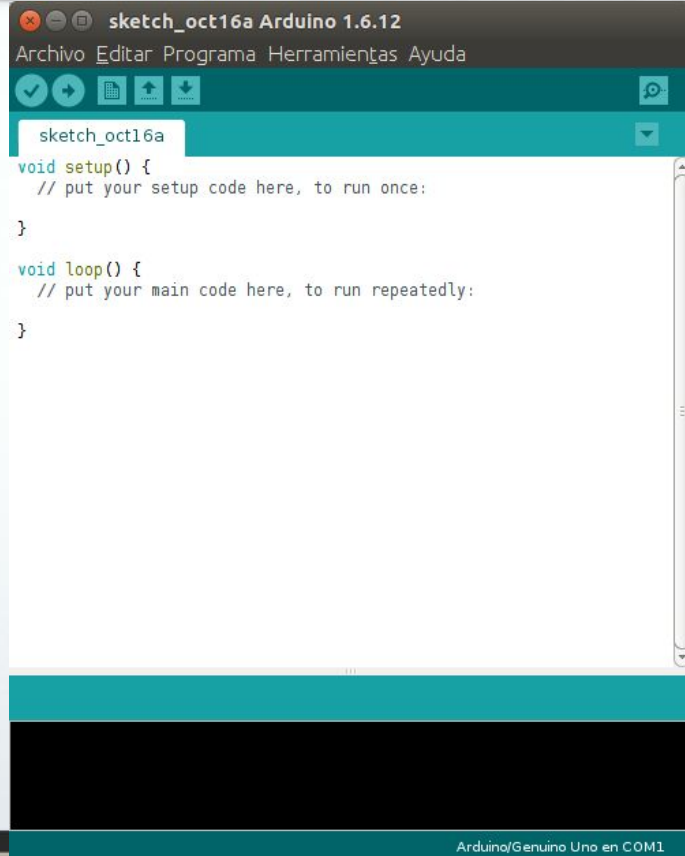


LINUX

1. Entramos en www.arduino.cc/linux y seleccionamos en www.arduino.cc/en/Main/Software la versión Linux.
2. Copiamos todos los archivos a un directorio y ejecutamos el script de Arduino.

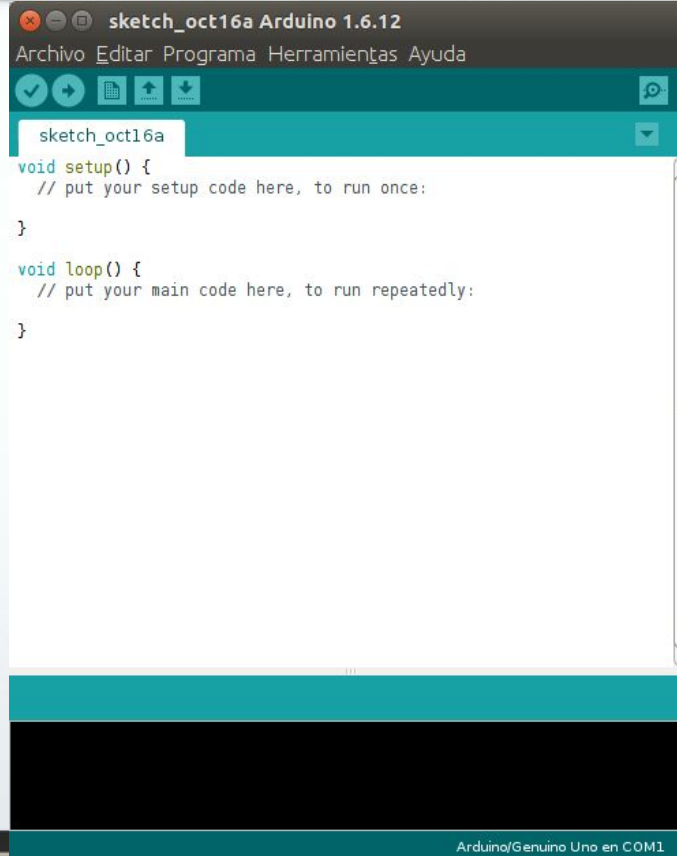


CONOCIENDO SOFTWARE ARDUINO









ZONA PARA ESCRIBIR
EL CÓDIGO →

ZONA DE MENSAJES
DEL SOFTWARE
(ERRORES Y ACCIONES) →



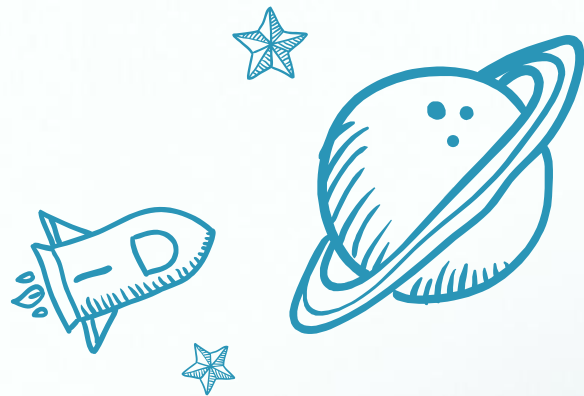
BARRA HERRAMIENTAS

Icono de la barra de herramientas	Acción
	Crea un nuevo sketch.
	Guarda el sketch.
	Abre un sketch.
	Verifica que el sketch este bien escrito.
	Carga un sketch a la tarjeta controladora.
	Inicia monitoreo del puerto serie.



En la parte superior tenemos el menú archivo con menús desplegables llamados “Archivo”, “Editar”, “Sketch”, “Herramientas” y “Ayuda”.

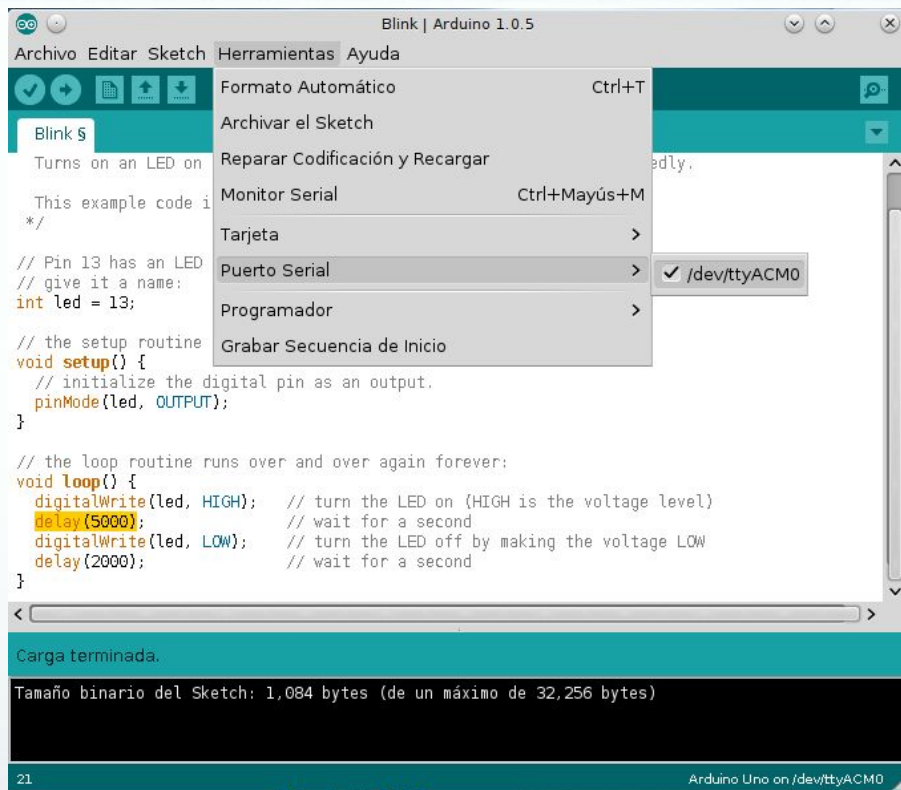




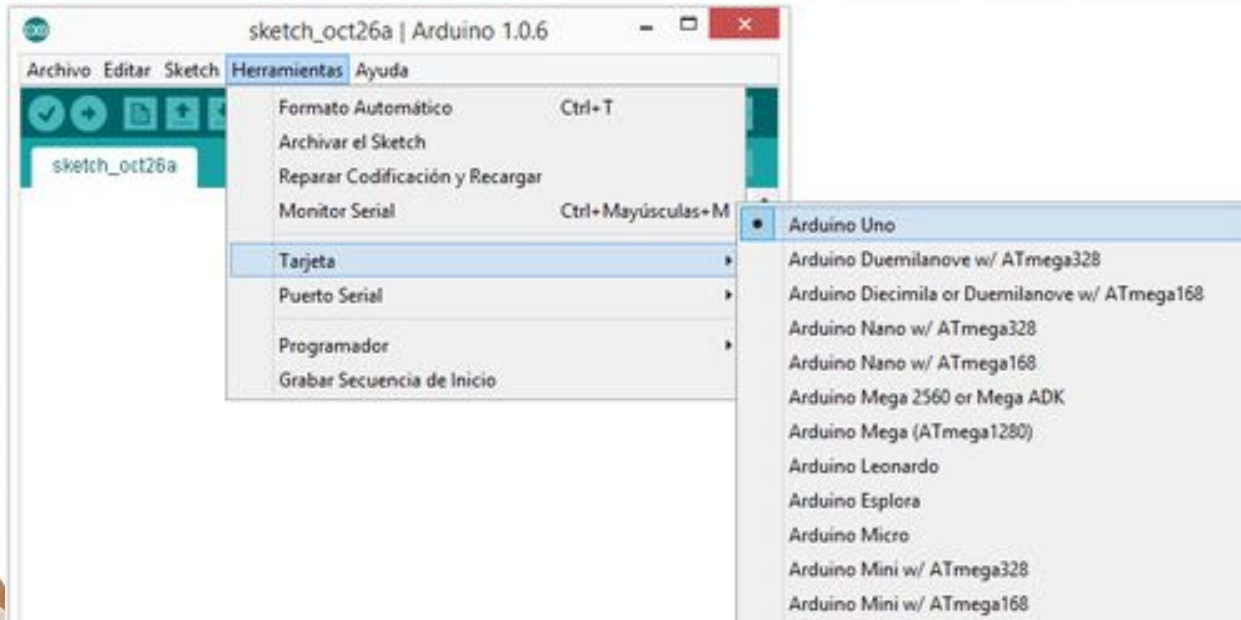
**VAMOS A CARGAR EL
PRIMER EJEMPLO**



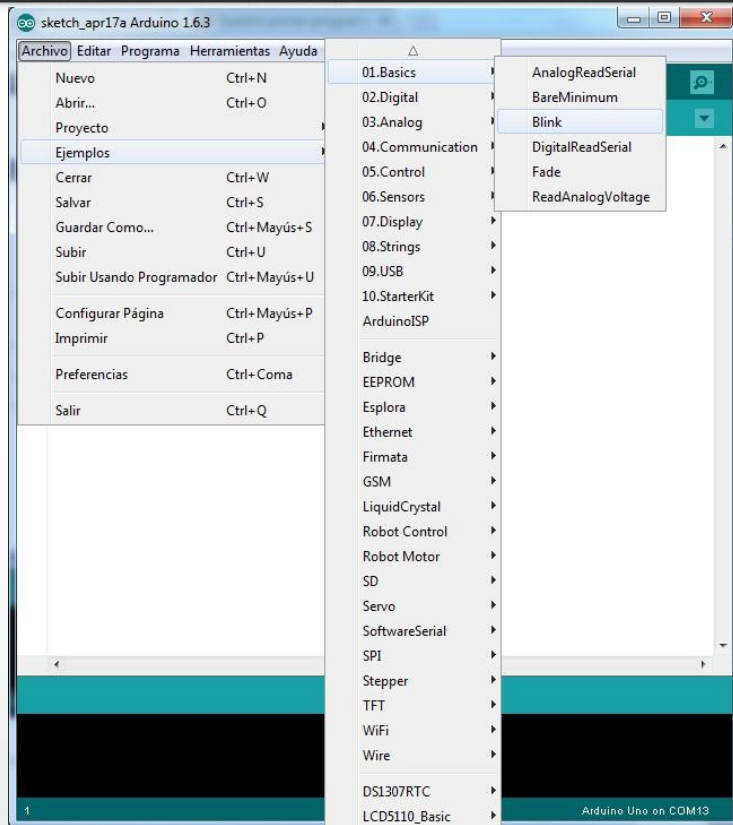
Conectamos el
Arduino al Pc y
seleccionamos el
puerto Com de la
placa.

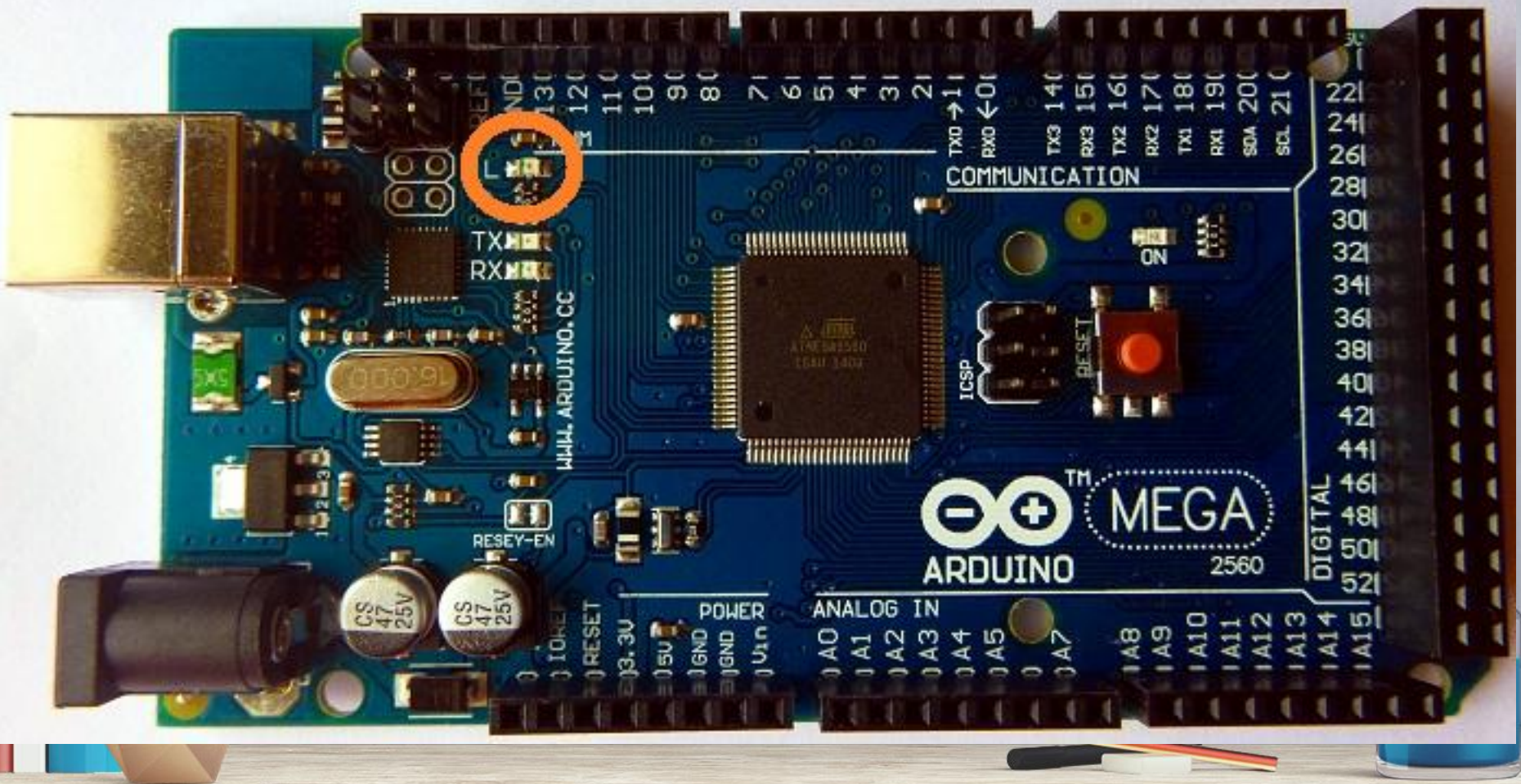


Seleccionamos la placa que tenemos



Cargamos nuestro primer ejemplo





El lenguaje en que se programan los Sketchs (como son llamadas las páginas de código para Arduino) es C++.

<http://arduino.cc/en/Reference/HomePage>





PROGRAMANDO CON ARDUINO



ESTRUCTURA BÁSICA DE PROGRAMACIÓN

Se divide en dos partes: `setup()` y `loop()`

- La función `setup()` constituye la declaración del programa.
- La función `loop()` es la ejecución.



- En la función **Setup()** se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa.
- Se ejecuta una vez y se usa para configurar las funcionalidades de la tarjeta Arduino (por ejemplo, si un puerto digital es de entrada o salida)



- La función `loop()` incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa)



**`/*Estructura básica de programación.
Funciones principales de Arduino*/`**

```
void setup() {  
    // put your setup code here, to run once:  
    //Preparación del programa y configuraciones  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    //Ejecución del programa principal  
}
```



- Toda instrucción o sentencia acaba con un un punto y coma (;)
- Una sentencia en su forma más simple es solo punto y coma.



- **Para hacer un comentario**
`// Comentario`
- **Bloque de comentario**
`*/ texto a comentar */`



Variables (Se declaran y se define el tipo de dato).

TIPO	NOMBRE	VALOR
boolean	mi_booleana	= false;
byte	mi_byte	= 10;
int	mi_int	= 1000;
float	mi_float	= 3,14;



- Variables globales : Se pueden usar en cualquier función del programa
- Se declaran antes de “void setup ()”

```
int v = 10;  
void setup () {  
}
```



- Variables locales: Solo se pueden usar en una única función.
- Se declaran dentro de la función.

```
void setup () {  
    int a = 10; // Variable local  
}
```

FUNCIONES

`pinMode(puerto,modo) : pinMode(13,OUTPUT)`

`digitalWrite(puerto,valor) : digitalWrite(10,HIGH)`

`delay(ms) : delay(1000)`

`Serial.println(dato) : Imprime datos`



CODEBENDER



**ES UN ENTORNO DE DESARROLLO EN LA
NUBE. PERMITE GUARDAR, ABRIR,
COMPARTIR, CARGAR, ETC ... PROYECTOS
EN LA NUBE Y TRABAJAR CON
PROYECTOS LOCALES**





FRITZING



fritzing

ES UN SOFTWARE PARA DISEÑAR LOS MONTAJES
EN PROTOBOARD Y A PARTIR DE ELLO GENERAR EL
PLANO Y LA PCB (CIRCUITO IMPRESO)

PERMITE DOCUMENTAR DISEÑOS Y PROTOTIPOS Y
CREAR ESQUEMAS DE CIRCUITOS IMPRESOS PARA
POSTERIOR FABRICACIÓN

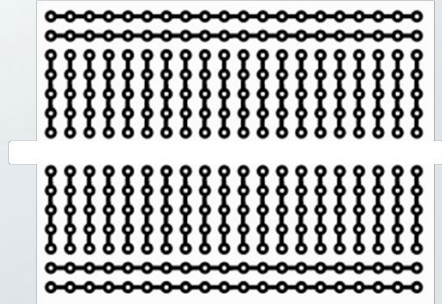
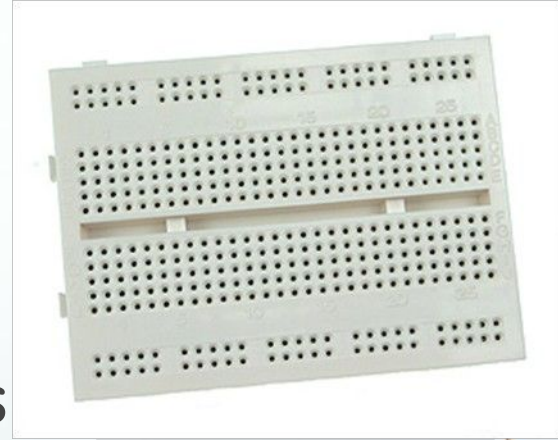


MATERIALES NECESARIOS PARA NUESTRA PRÁCTICA CON ARDUINO



PROTOBOARD

Placa para construir circuitos sin soldar. Compuesta por bloques de plástico perforados y láminas de una aleación de cobre, estaño y fósforo.



RESISTENCIA

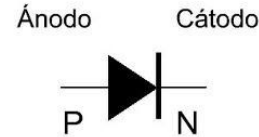


Formado por carbón y otros elementos resistivos para disminuir la corriente que pasa.

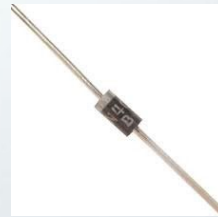
Se identifica por las bandas de colores. (0,25W–0,5W y 1W)



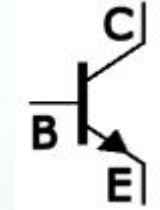
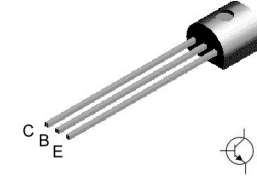
DIODO



Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido. Tiene dos partes: el cátodo y el ánodo.



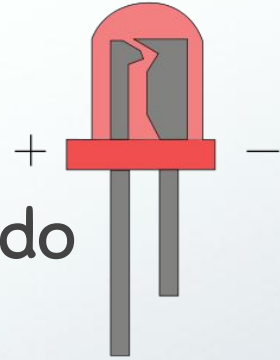
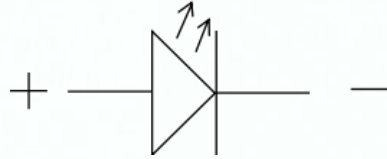
TRANSISTOR



El transistor es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador. Tiene tres partes: la base (B), el emisor (E) y colector (C).



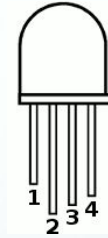
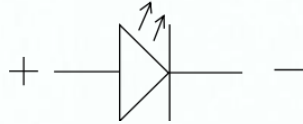
LED



Un LED (Diodo emisor de luz, también "diodo luminoso") es un diodo semiconductor que emite luz.

El LED tiene una polaridad, un orden de conexión, y al conectarlo al revés se puede quemar

LED RGB

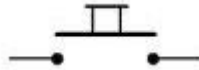


RGB LED

- 1: Green (+)
- 2: Ground (-)
- 3: Blue (+)
- 4: Red (+)

Un LED RGB es un LED que incorpora en su mismo encapsulado tres LEDs, es RGB porque R (red, rojo), G (green, verde) y B (blue, azul) así se pueden formar miles de colores ajustando de manera individual cada color. Los tres LEDs están unidos por el negativo o cátodo.

PULSADOR



Un botón funciona como un interrupto eléctrico, en su interior tiene dos contactos, si es un dispositivo NA (normalmente abierto) o NC (normalmente cerrado), con lo que al pulsarlo se activará la función inversa de la que en ese momento esté realizando.



POTENCIOMETRO

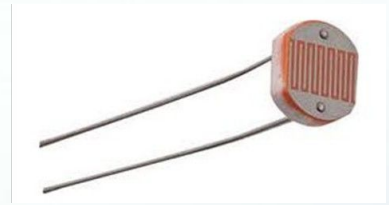
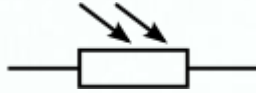


Un potenciómetro es una resistencia cuyo valor de resistencia es variable.

De esta manera, indirectamente, se puede controlar la intensidad de corriente que fluye por un circuito si se conecta en paralelo, o controlar el voltaje al conectarlo en serie.



FOTOCELULA



Es una resistencia, cuyo valor en ohmios varía ante las variaciones de la luz incidente.

Una fotocelda presenta un bajo valor de su resistencia ante la presencia de luz y un alto valor de resistencia ante la ausencia de luz.



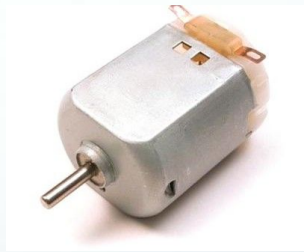
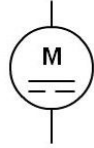
BUZZER



2 elementos, un electroimán y una lámina metálica de acero. El zumbador puede ser conectado a circuitos integrados especiales para así lograr distintos tonos. Cuando se acciona, la corriente pasa por la bobina del electroimán y produce un campo magnético variable que hace vibrar la lámina de acero sobre la armadura.



MOTOR DC



El motor de corriente continua (DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio.



SERVOMOTOR

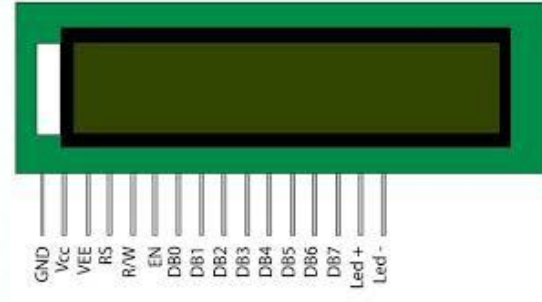


Dispositivo similar a un motor de corriente continua que puede ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Un servomotor es un motor eléctrico que puede ser controlado tanto en velocidad como en posición.



PANTALLA LCD



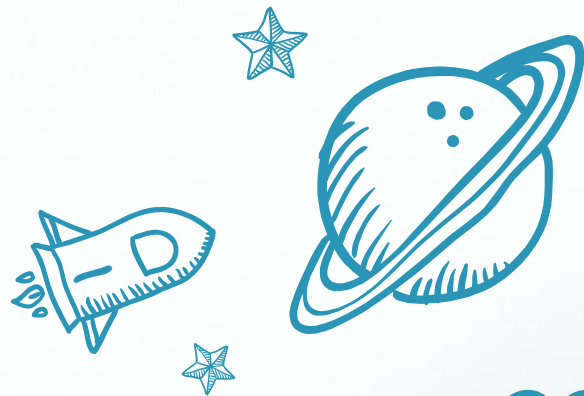
Un display LCD (Liquid Crystal Display) es un display alfanumérico de matriz de puntos (dot-matrix) que sirve para mostrar mensajes a través de caracteres como letras, números o símbolos.

PUENTE H L293D



Es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar, retroceder.





EMPEZAMOS???

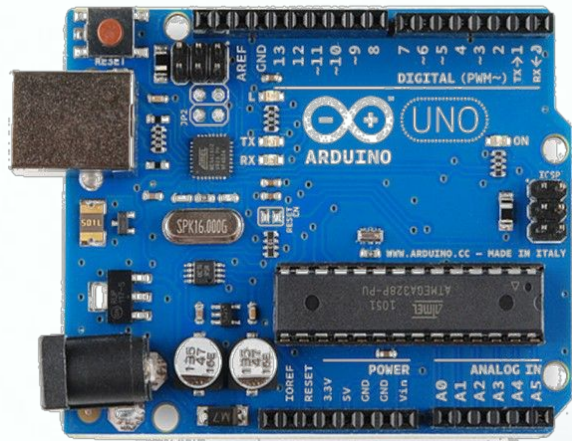


LED INTERMITENTE 1

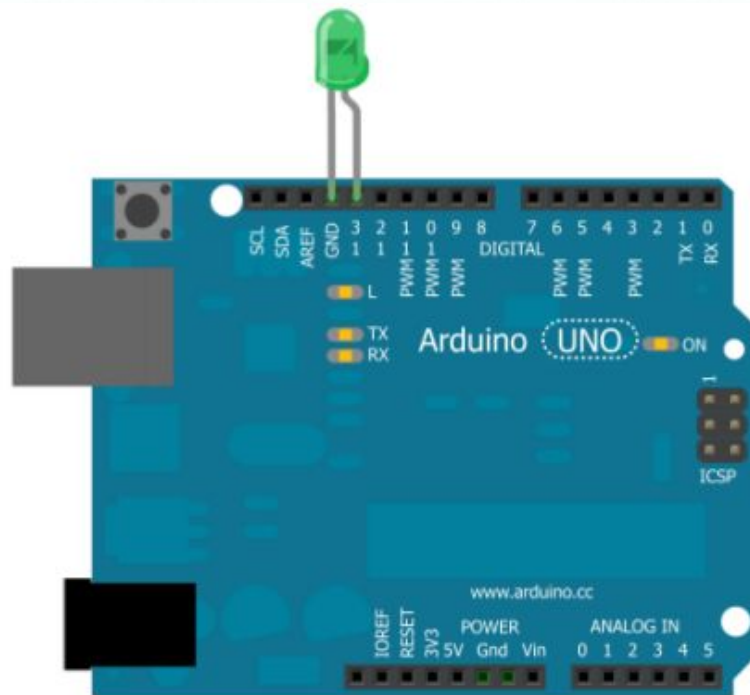
En este ejercicio vamos a encender un led conectado a nuestra placa arduino.



LED INTERMITENTE 1



LED INTERMITTENTE 1



CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
ENCIENDELED1.INO DENTRO DE LA CARPETA 2.

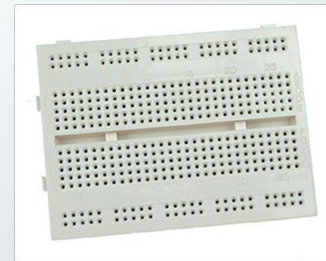
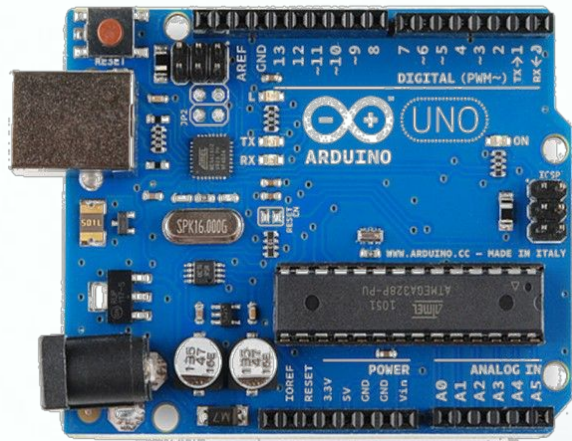


LED INTERMITENTE 2

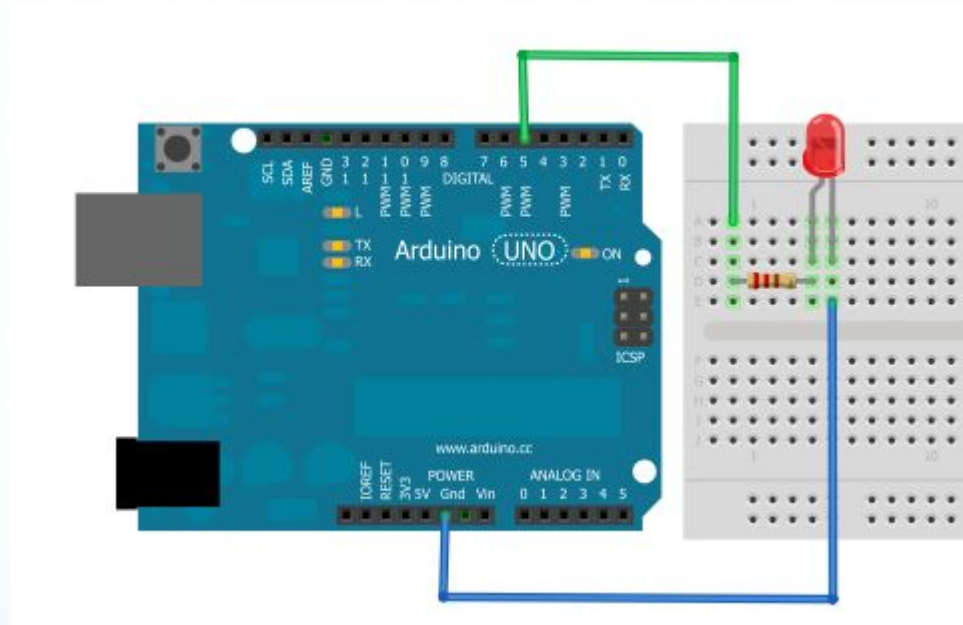
En este ejercicio vamos a encender un led conectado a nuestra placa y con una resistencia usando la protoboard.



LED INTERMITENTE 2



LED INTERMITTENTE 2



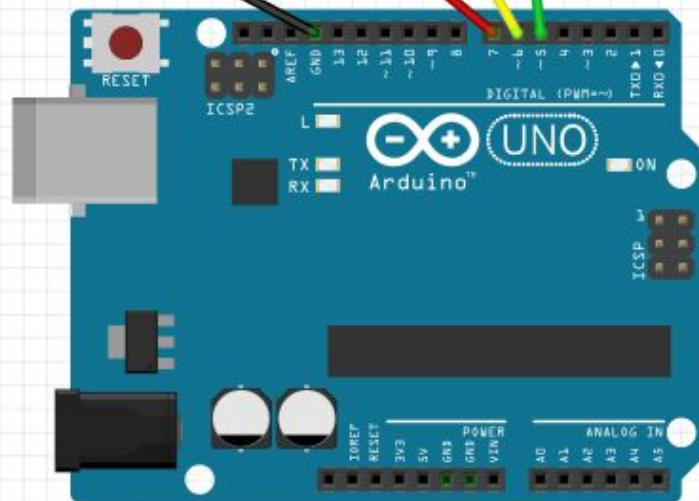
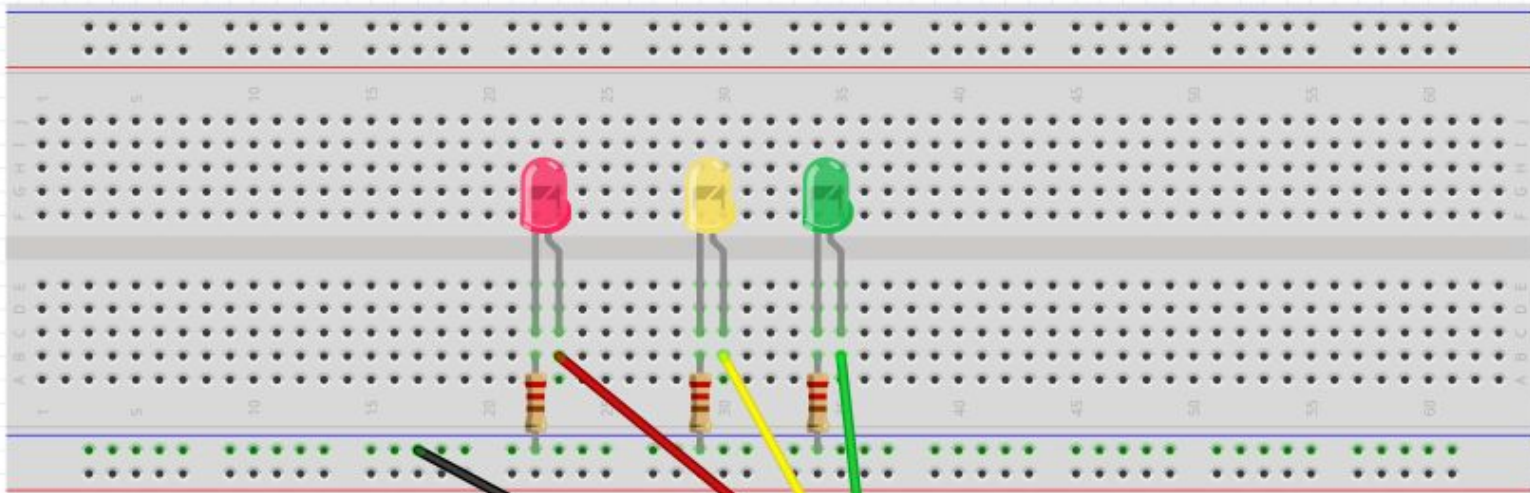
CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
ENCIENDELED5.INO DENTRO DE LA CARPETA 3.



VARIOS LEDS INTERMITENTES

El reto consiste en encender y apagar al menos 3 leds



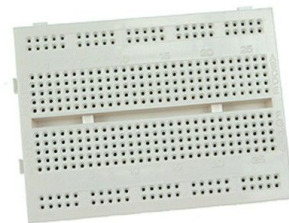
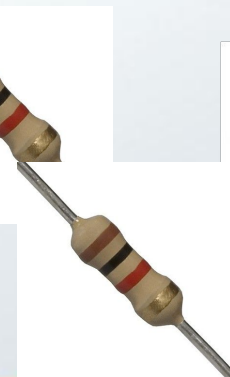
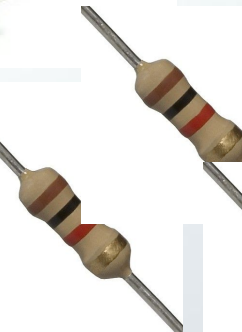
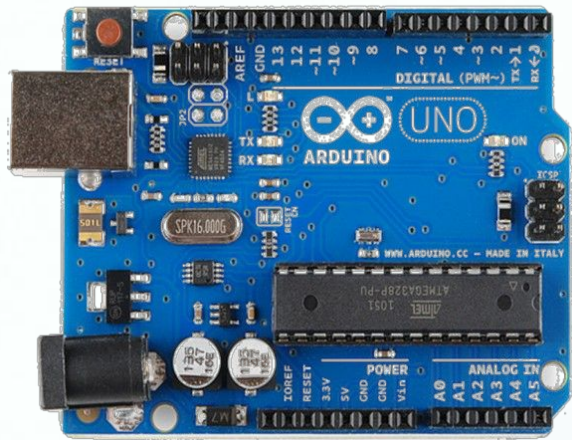


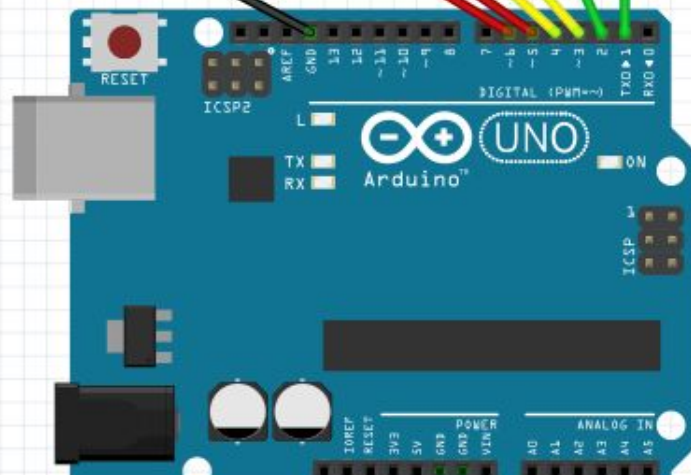
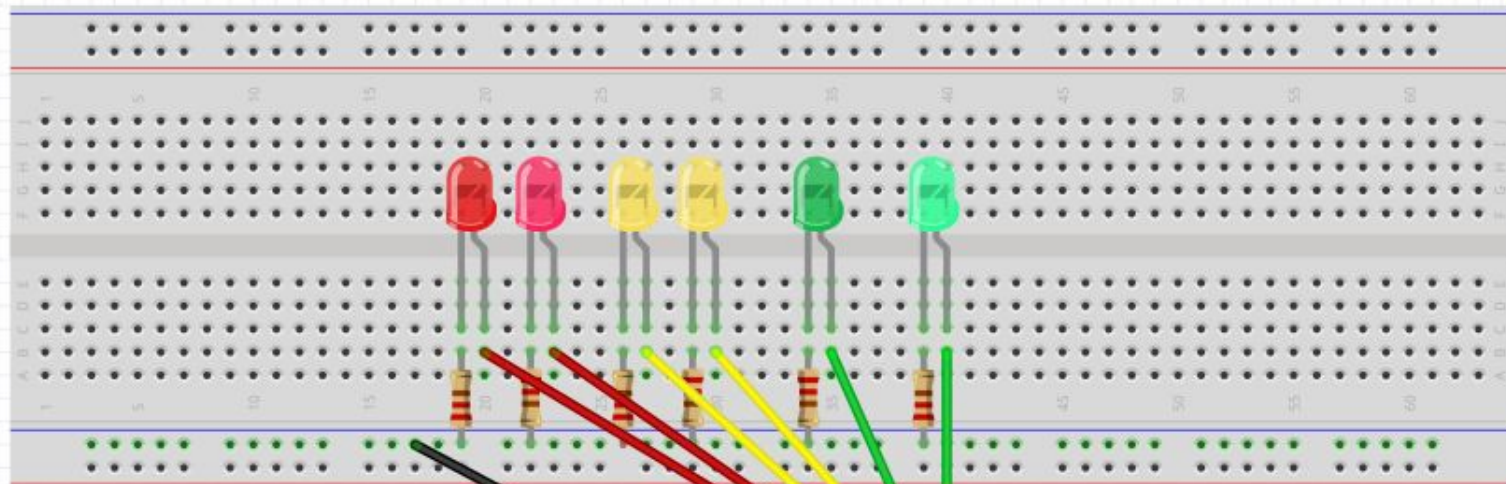
fritzing

CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
ENCIENDE VARIOS LEDS.INO DENTRO DE LA CARPETA 4



COCHE FANTASTICO





fritzing



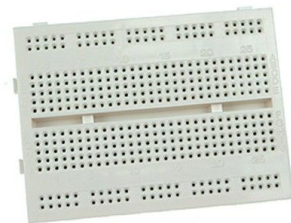
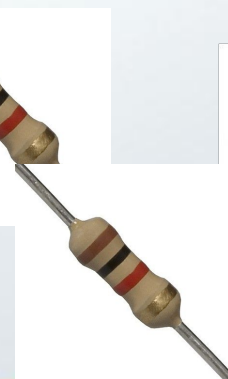
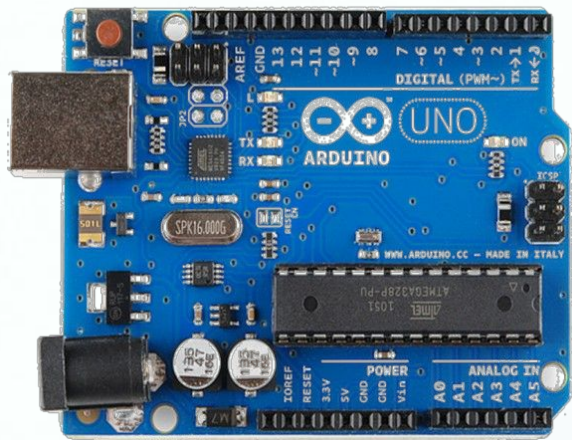
CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
COCHFANTASTICOFACIL.INO DENTRO DE LA CARPETA
5



CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
COCHFANTASTICO2.INO



SEMAFORO DE CRUCE



SEMAFORO DE CRUCE

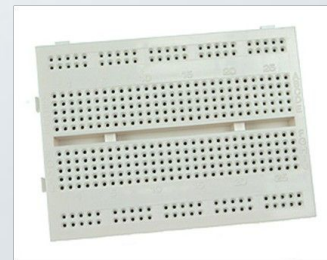
Necesitamos 4 leds verde, 4 rojo y dos amarillos.

C= Coche P=Peatón V=Verde R=Rojo A=Amarillo

1° C=V; P=R -- C=R; P=V 2° C=A; P=R -- C=R; P=V

3° C=R -- P=V C=A -- P=R 4° C=A P=R C=R P=V

LED CON PULSADOR



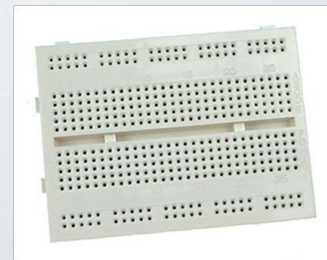
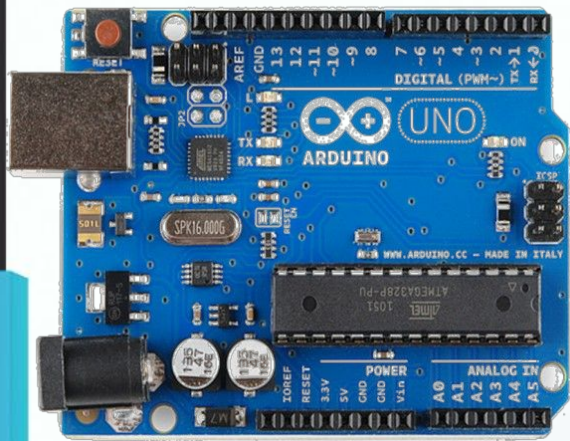


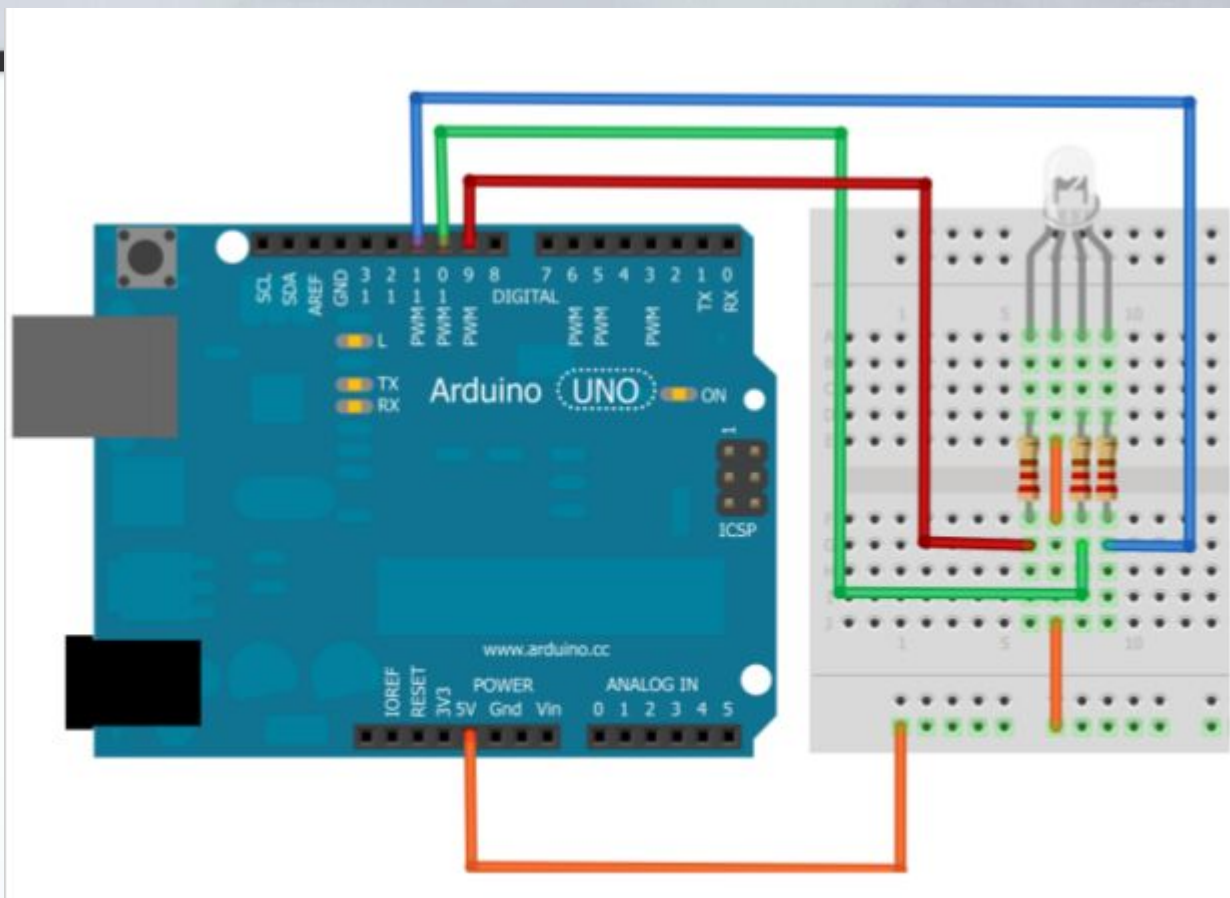
CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
INTERRUPTORLED.INO EN CARPETA 8 Y 9

[HTTPS://OPENWEBINARS.NET/TUTORIAL-ARDUINO-ENTRADAS-2-BOTONES/](https://openwebinars.net/tutorial-arduino-entradas-2-botones/)



LED RGB





**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
RGB_SENCILLO EN CARPETA 10**



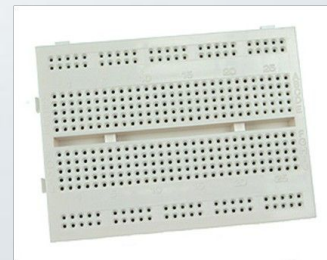
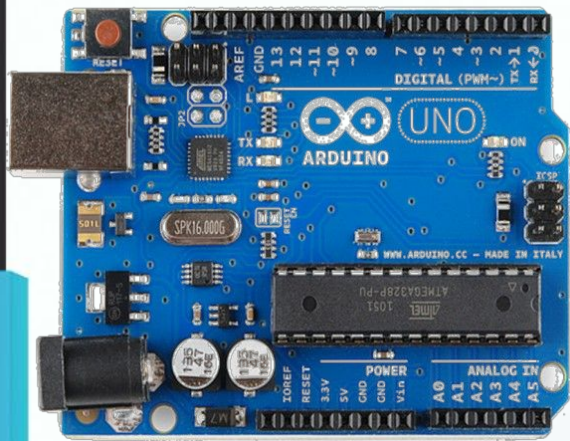
PODEMOS PROBAR CON LEDRGB.INO EN CARPETA 10

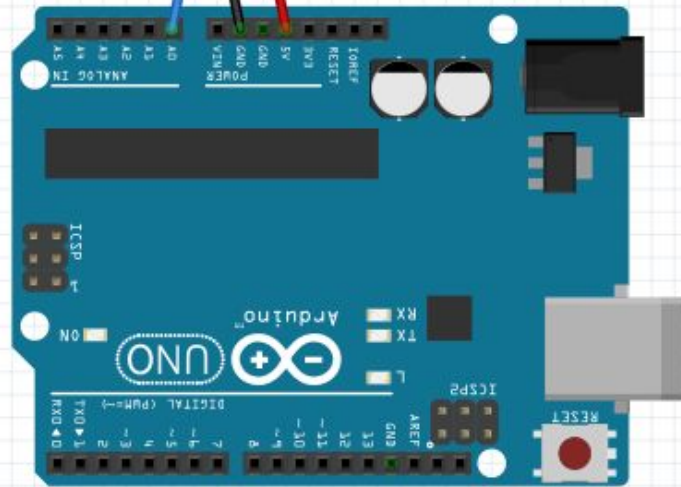
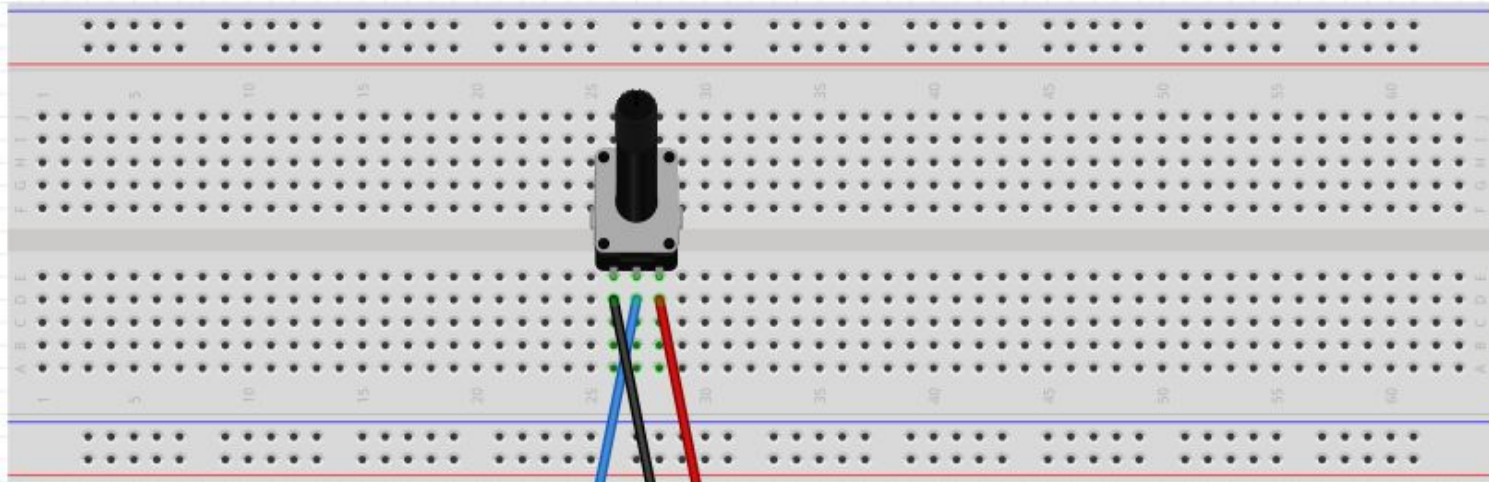


**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
LEDRGB_RANDOM.INO EN CARPETA 11**



POTENCIOMETRO



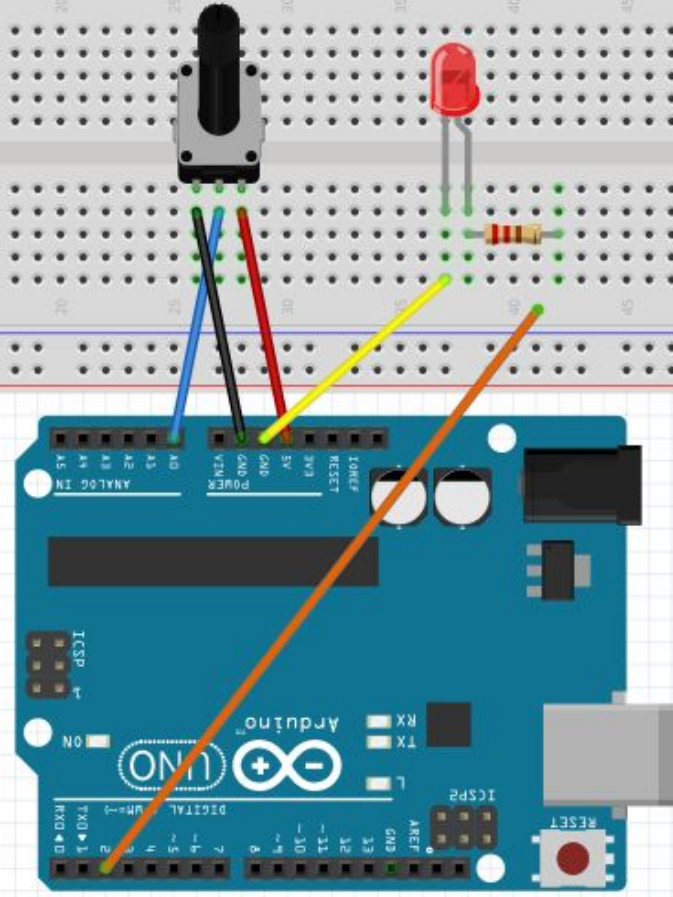


fritzing

**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
POTENCIOMETROLEE.INO EN CARPETA 12**



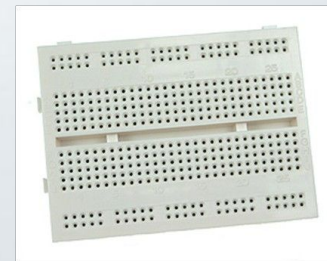
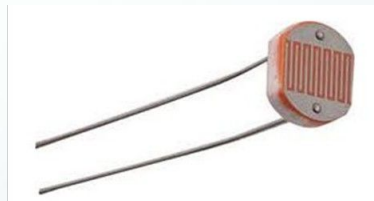
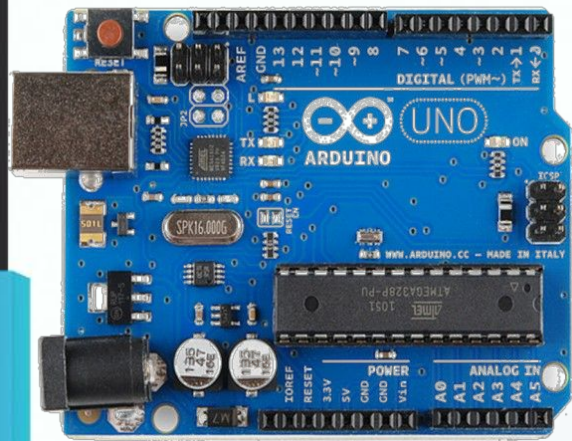
fritzing

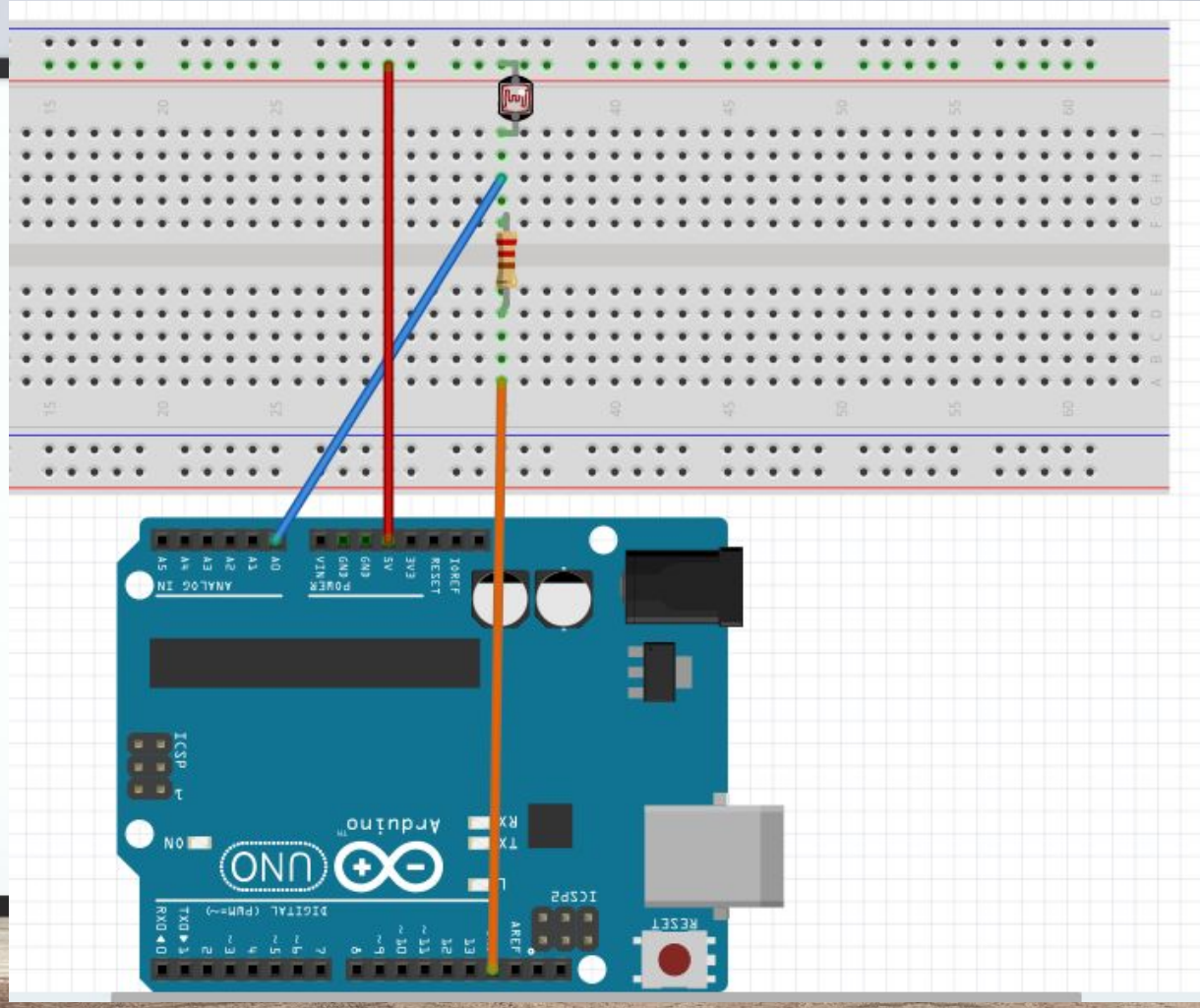


**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
LEDPOTENCIOMETRO.INO EN CARPETA 13**



SENSOR DE LUZ

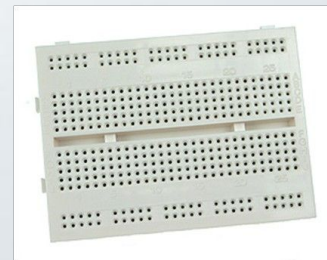
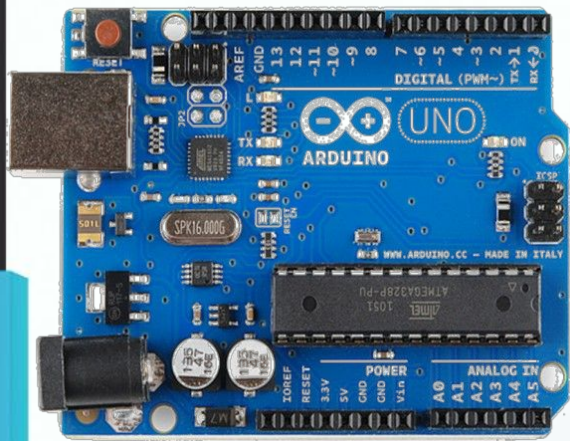


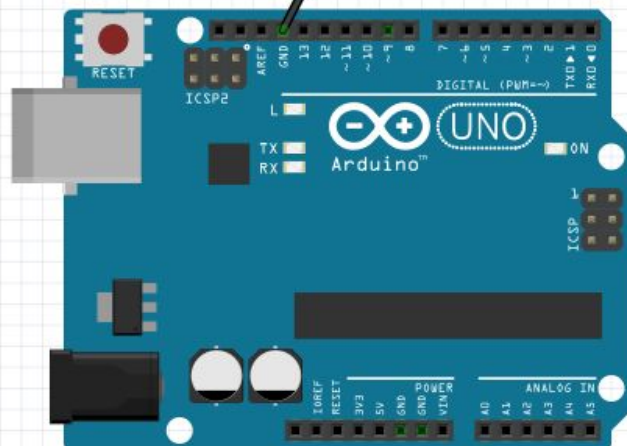
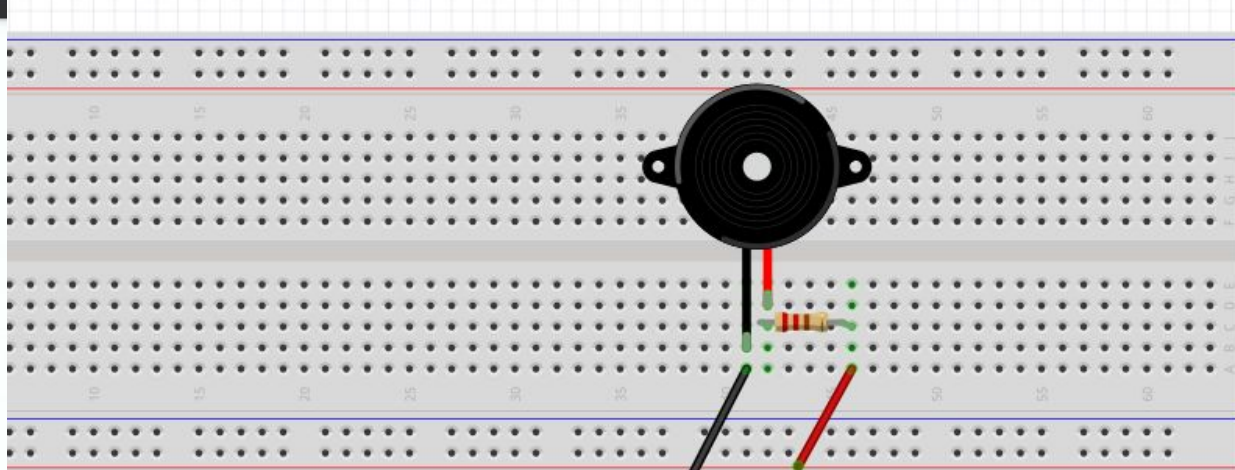


**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
SENSORLUZLDR.INO EN CARPETA 15**



SPEAKER





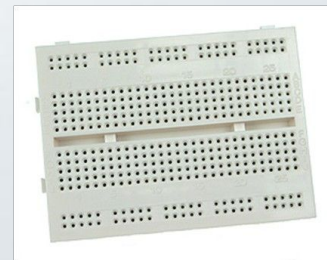
itizing

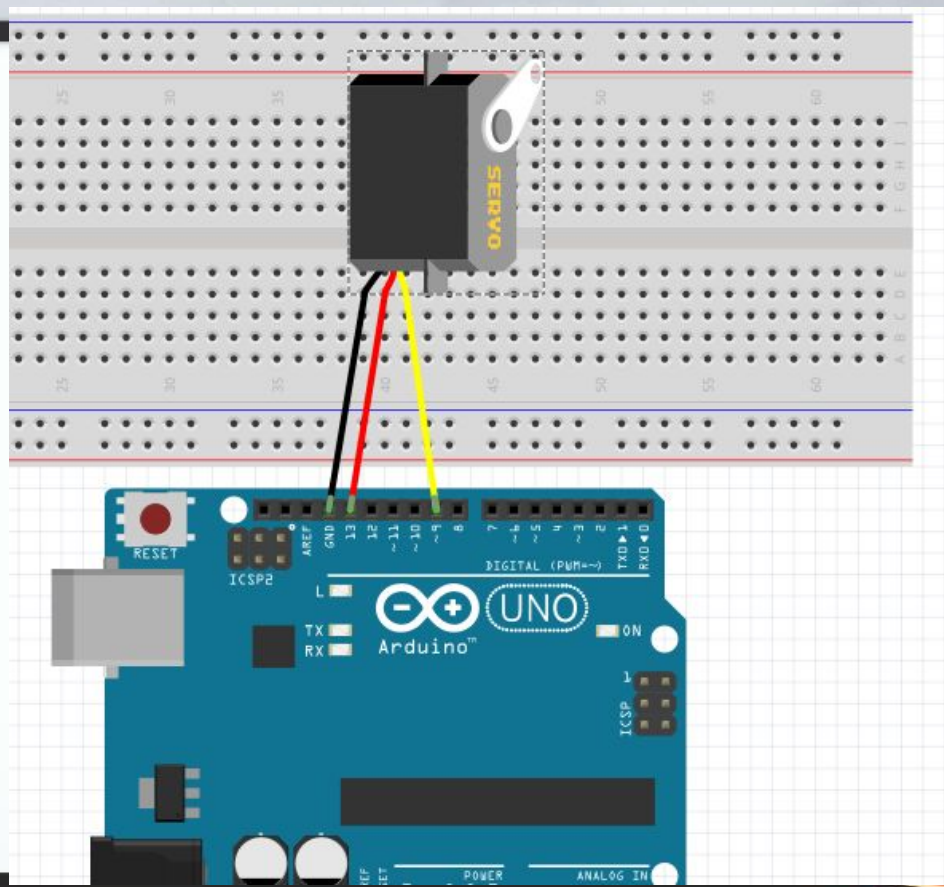


**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
SPEAKER.INO**



SERVO





**CARGAMOS EN EL IDE DE ARDUINO EN EJEMPLO LLAMADO
SERVO.INO**

