



ES UNA PLATAFORMA DE PROTOTIPO ELECTRONICA DE DISEÑO ABIERTO. ¿QUE SIGNIFICA ESTO?

ABIERTO-

"RECURSOS QUE PUEDEN SER USADOS, REDISTRIBUIDOS O MODIFICADOS DE FORMA GRATUITA. A MENUDO SOFTWARE O HARDWARE."

ELECTRONICA-

"TECNOLOGIA QUE HACE USO DEL MOVIMIENTO CONTROLADO DE ELECTRONES A TRAVES DE DIFERENTES MEDIOS."

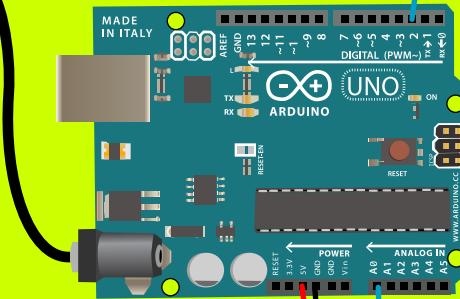
PROTOTIPO-

"UNA FORMA ORIGINAL QUE PUEDE SERVIR COMO BASEO STANDAR PARA OTROS OBJETOS."

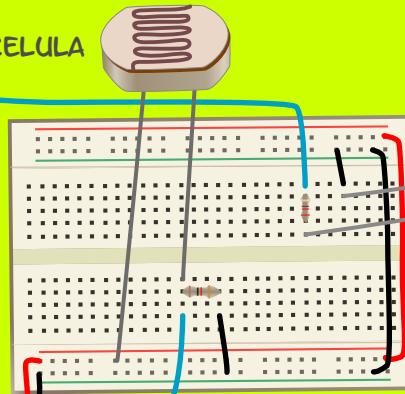
PLATAFORMA-

"ARQUITECTURA DE HARDWARE CON UNA BASE DE SOFTWARE QUE LE PERMITE EJECUTAR PROGRAMAS."

MICROCHIP



FOTOCELULA



LED



PLACA DE PROTOTIPO

UN ARDUINO CONTIENE UN MICROCHIP, QUE ES UN ORDENADOR MUY PEQUEÑO QUE PODEMOS PROGRAMAR. SE LE PUEDEN CONECTAR SENSORES PARA QUE PUEDA MEDIR CONDICIONES (COMO CUANTO LUZ HAY EN LA HABITACIÓN). PUEDE CONTROLAR OTROS OBJETOS PARA QUE REACCIONEN A ESAS CONDICIONES (SI LA HABITACIÓN ESTÁ A OSCURAS ENCIEDE EL LED).

ENCENDIDO

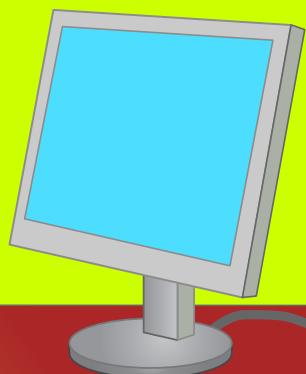


APAGADO

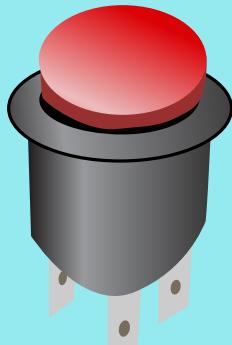


SE PUEDE REACCIONAR CUANDO UN INTERRUPTOR ES ACCIONADO.

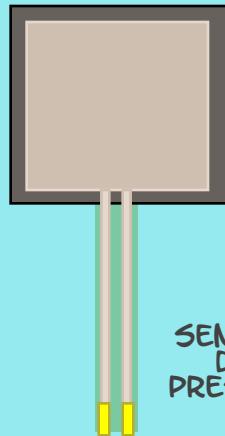
UN RATÓN ES UN DISPOSITIVO DE ENTRADA COMÚN PARA UN ORDENADOR, UN MONITOR ES UN DISPOSITIVO DE SALIDA.



LOS MICROCONTROLADORES USAN ENTRADAS Y SALIDAS COMO CUALQUIER ORDENADOR. LAS ENTRADAS LEEN INFORMACIÓN DEL USUARIO O EL ENTORNO MIENTRAS QUE LAS SALIDAS HACEN ALGO CON LA INFORMACIÓN QUE HA SIDO LEIDA.



PULSADOR

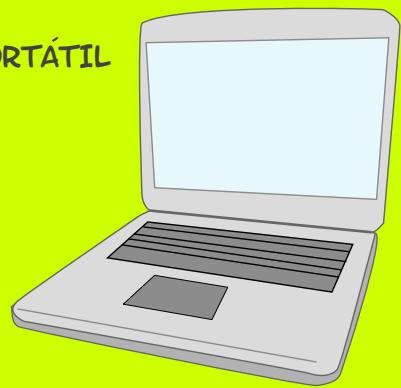


SENSOR  
DE  
PRESIÓN

UNA ENTRADA PARA ARDUINO PUEDE SER UN PULSADOR O UN SENSOR.



MOTOR DC



PORTÁTIL

CUALQUIER OBJETO QUE QUERAMOS ENCENDER, APAGAR Y CONTROLAR PUEDE SER UNA SALIDA. PUEDE SER UN MOTOR O INCLUSO UN ORDENADOR.



¿CUAL ES LA DIFERENCIA ENTRE LAS ENTRADAS Y SALIDAS DIGITALES O ANALÓGICAS?

LAS ENTRADAS Y SALIDAS PUEDEN SER DIGITALES O ANALÓGICAS. EN LAS DIGITALES LA INFORMACIÓN ES BINARIA - SOLO PUEDE SER VERDADERO O FALSO. LA INFORMACIÓN ANALÓGICA ES CONTINUA, PUEDE MANEJAR UN AMPLIO RANGO DE VALORES.

LA INFORMACIÓN DIGITAL ES DISCRETA Y FINITA. TODA LA INFORMACIÓN SE DESCRIBE CON DOS ESTADOS, 1 O 0, ENCENDIDO O APAGADO.

LA INFORMACIÓN ANALÓGICA SE CARACTERIZA POR SU NATURALEZA CONTINUA. PUEDE SER UN NÚMERO INFINITO DE VALORES POSIBLES.



UN PULSADOR ES UNA ENTRADA DIGITAL, UN SENSOR ES UNA ENTRADA ANALÓGICA. EL RANGO DE UN SENSOR ESTÁ LIMITADO POR SU CONVERSIÓN A DATO DIGITAL.



¿VOLTAJE?  
¿INTENSIDAD?  
¿RESISTENCIA?  
¿LEY DE OHM?

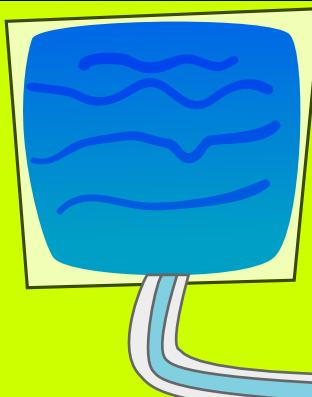
**VOLTAJE (V)**  
ES LA MEDIDA DEL POTENCIAL ELÉCTRICO EN UN CIRCUITO. SE MIDE EN **VOLTIOS**.

**INTENSIDAD (I)**  
ES LA CANTIDAD DE FLUJO QUE ATRAVIESA UN MATERIAL CONDUCTOR. SE MIDE EN **AMPERIOS O AMPS**.

**RESISTENCIA (R)**  
ES LA OPOSICIÓN DEL MATERIAL AL FLUJO DE LA CORRIENTE ELÉCTRICA. SE MIDE EN **OHMIOS**.

ANTES DE CONECTAR EL ARDUINO, DEBEMOS REPASAR UNOS POCOS TÉRMINOS Y PRINCIPIOS SOBRE COMO LA ELECTRICIDAD (Y LA ELECTRÓNICA) FUNCIONA.

LA ELECTRICIDAD ES EL FLUJO DE LA ENERGÍA ELÉCTRICA A TRAVÉS DE UN MATERIAL CONDUCTOR.



LA VELOCIDAD DE FLUJO ESTÁ DETERMINADA POR EL VOLTAJE

LA RESISTENCIA AUMENTA O DISMINUYE EL FLUJO



LA CANTIDAD DE AGUA QUE SE MUEVE POR LA MANGUERA ES LA INTENSIDAD

LA ANALOGÍA DEL AGUA SE USA NORMALMENTE PARA EXPLICAR ESTOS TÉRMINOS. AQUÍ HAY UN MODELO.

## LEY DE OHM

INTENSIDAD = VOLTAJE/RESISTENCIA

$$(I = V/R)$$

O

RESISTENCIA = VOLTAJE/INTENSIDAD

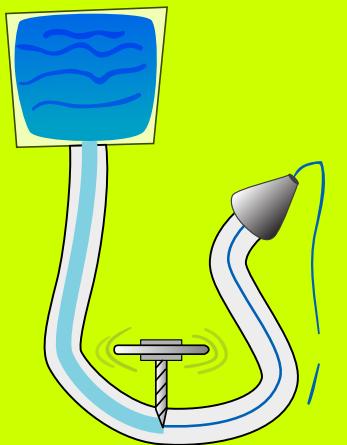
$$(R = V/I)$$

O

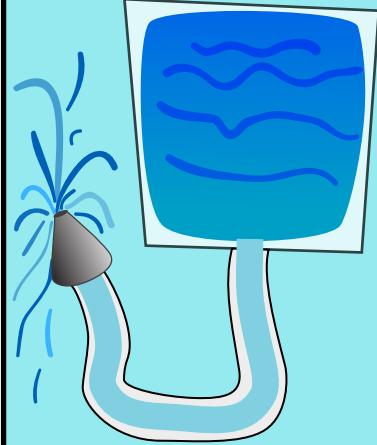
VOLTAJE = RESISTENCIA \* INTENSIDAD

$$(V = R*I)$$

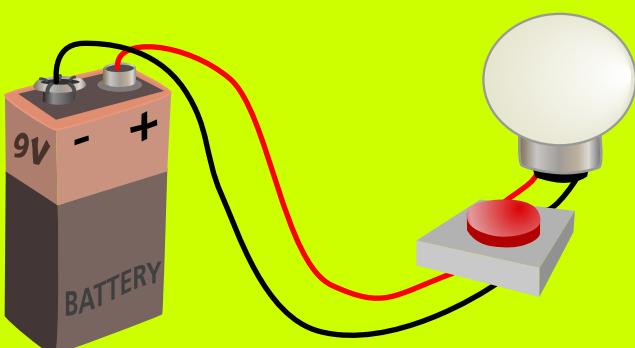
HAY UNA RELACION ENTRE VOLTAJE, INTENSIDAD Y RESISTENCIA DESCUBIERTA POR GEORG OHM, UN FISICO ALEMAN.



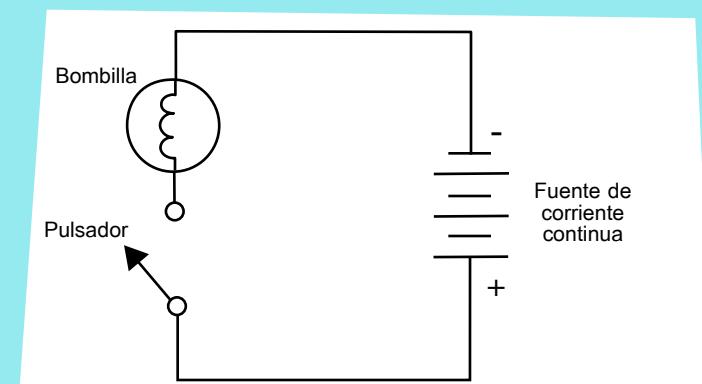
POR EJEMPLO  
INCREMENTANDO LA  
RESISTENCIA  
SE REDUCE EL FLUJO.



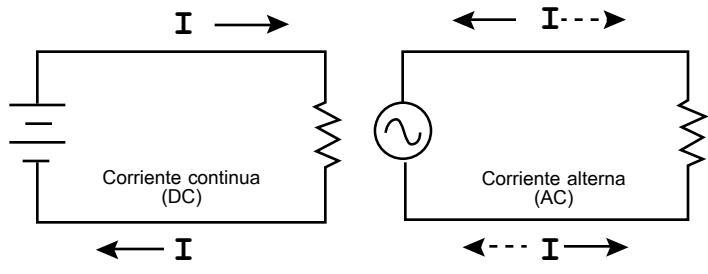
O INCREMENTANDO EL  
POTENCIAL, MAS FLUJO.



AHORA MIREMOS UN CIRCUITO SENCILLO. CADA CIRCUITO ES UN CIRCULO CERRADO QUE TIENE UNA FUENTE DE ENERGIA (BATERIA) Y UNA CARGA (BOMBILLA). LA CARGA TRANSFORMA LA ENERGIA ELECTRICA DE LA BATERIA Y LA UTILIZA. ESTE TAMBien TIENE UN PULSADOR.



ESTE ES EL ESQUEMA DEL MISMO CIRCUITO  
(REPRESENTA EL CIRCUITO UTILIZANDO SIMBOLOS  
PARA LOS COMPONENTES ELECTRONICOS). CUANDO EL  
PULSADOR SE CIERRA LA CORRIENTE FLUYE DESDE LA  
FUENTE DE ENERGIA Y ENCIENDE LA BOMBILLA.



HAY DOS TIPOS DE CIRCUITOS, LOS DE CORRIENTE CONTINUA Y LOS DE CORRIENTE ALTERNA. EN LOS CIRCUITOS DC LA CORRIENTE SIEMPRE FLUYE EN UNA DIRECCIÓN, EN LOS DE AC LA CORRIENTE FLUYE EN DIRECCIONES OPUESTAS EN CICLOS REGULARES. NOSOTROS SOLO HABLAREMOS DE CIRCUITOS DC.



CONECTAR LA ARDUINO A UN ORDENADOR CON UN CABLE USB LE PROPORCIONA LA ENERGIA QUE NECESITA Y NOS PERMITE PROGRAMARLA.



TIENES QUE DESCARGAR E INSTALAR ESTE SOFTWARE PARA PROGRAMAR UNA ARDUINO, ESTA DISPONIBLE EN LA URL DE ARRIBA DE FORMA TOTALMENTE LIBRE. EL SOFTWARE DE ARDUINO FUNCIONA EN MAC OS X, WINDOWS Y LINUX.

PARA INSTRUCCIONES SOBRE COMO HACER LA INSTALACION DEL SOFTWARE DE ARDUINO EN UN MAC:

[HTTP://WWW.ARDUINO.CC/ES/GUIDE/MACOSX](http://www.arduino.cc/es/guide/macOSX)

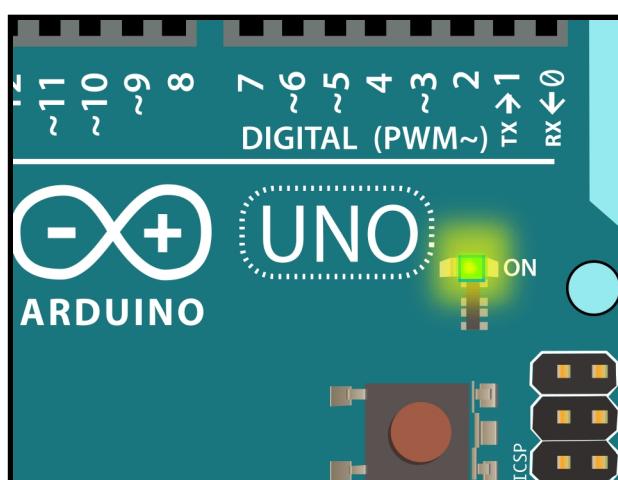
INSTRUCCIONES PARA LA INSTALACION EN WINDOWS:

[HTTP://WWW.ARDUINO.CC/ES/GUIDE/WINDOWS](http://www.arduino.cc/es/guide/windows)

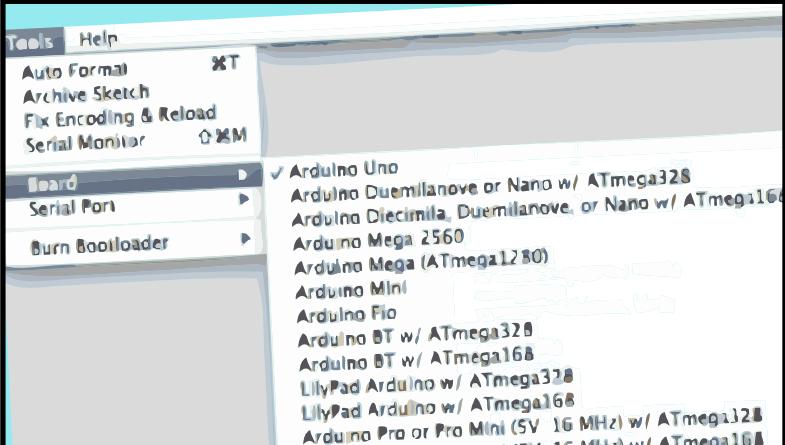
INTRUCCIONES PARA LA INSTALACION EN LINUX:

[HTTP://WWW.ARDUINO.CC/PLAYGROUND/LEARNING/LINUX](http://www.arduino.cc/playground/learning/linux)

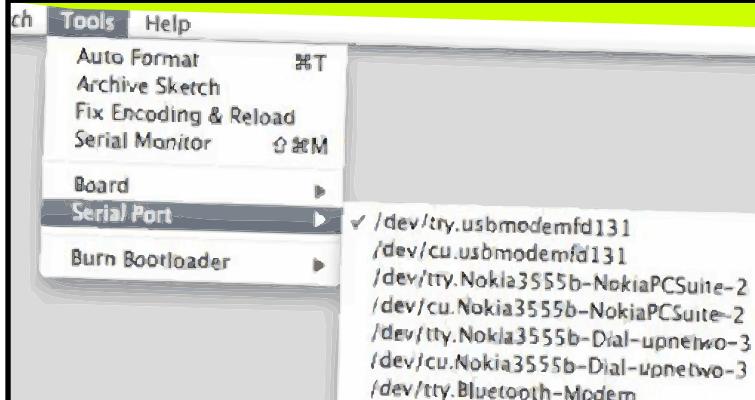
VE A LAS ANTERIORES URLs DE ARRIBA PARA OBTENER DETALLADAS INSTRUCCIONES SOBRE LA INSTALACION EN CADA UNA DE ESAS PLATAFORMAS.



CUANDO TENGAS INSTALADO EL SOFTWARE, CONECTA LA ARDUINO. UN LED MARCADO COMO ON DEBERIA ENCENDERSE EN LA PLACA.



LANZAMOS EL SOFTWARE DE ARDUINO. EN EL MENU TOOLS, SELECCIONA LA PLACA QUE ESTAS UTILIZANDO (TOOLS > BOARD). POR EJEMPLO, ARDUINO UNO.

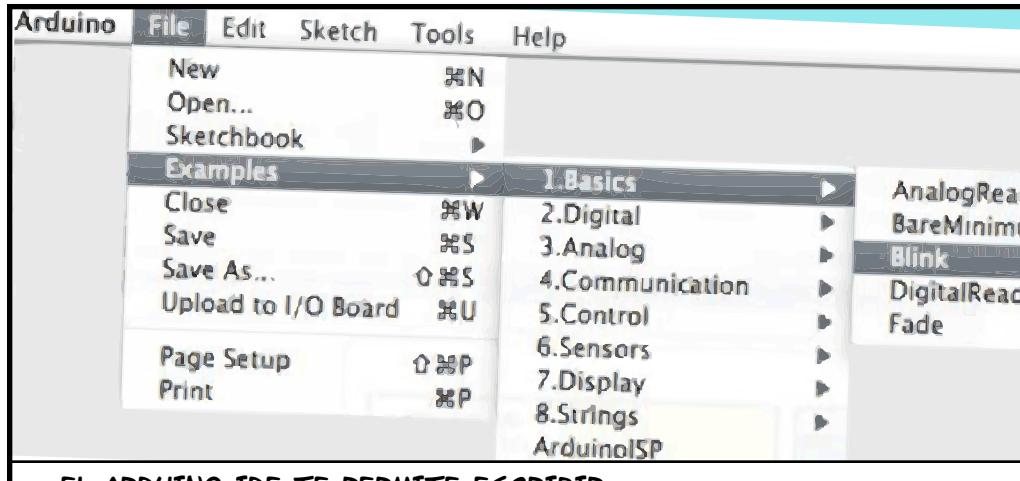


A CONTINUACION SELECCIONA EL PUERTO SERIE. (TOOLS > SERIAL PORT) EN UN MAC DEBE SER ALGO PARECIDO A /DEV/TTY.USBMODEM. EN UNA MAQUINA WINDOWS SERA COM3 O ALGO PARECIDO.

¿QUE ES UN  
ENTORNO DE  
DESARROLLO  
INTEGRADO  
(IDE)?



CUANDO DESCARGAS EL SOFTWARE DE ARDUINO DESCARGAS UN IDE. ESTE COMBINA UN EDITOR DE TEXTO CON UN COMPILEADOR Y OTRAS FUNCIONES PARA AYUDAR A LOS PROGRAMADORES A ESCRIBIR PROGRAMAS.

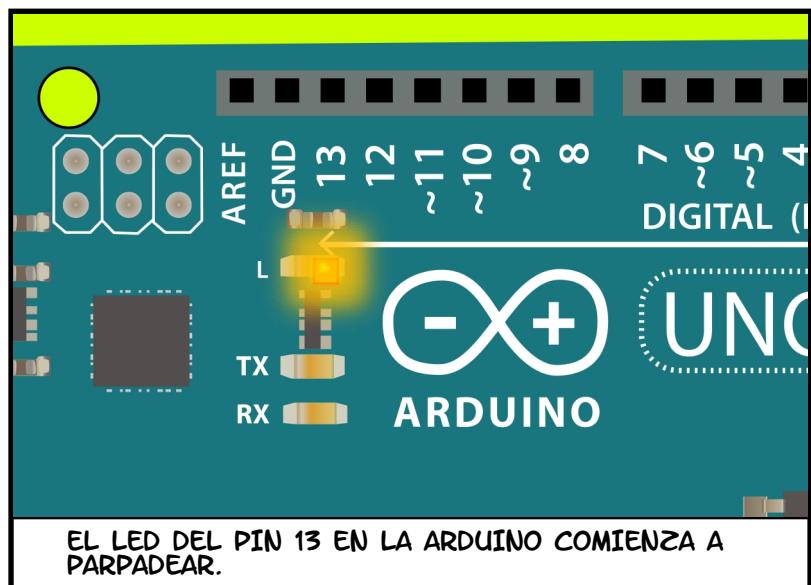


EL ARDUINO IDE TE PERMITE ESCRIBIR SKETCHES O PROGRAMAS Y SUBIRLOS A LA PLACA ARDUINO. ABRE EL EJEMPLO BLINK EN EL MENU FILE. FILE > EXAMPLES > 1.BASICS > BLINK.

BOTON PARA SUBIR

```
int ledPin = 13;  
  
void setup() {  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(ledPin, HIGH);  
}
```

PARA SUBIR UN SKETCH A LA PLACA ARDUINO PINCHAMOS EN EL BOTON DE SUBIR EN LA TIRA DE BOTONES EN LA PARTE SUPERIOR DE LA VENTANA. APARECERAN MENSAJES EN LA PARTE INFERIOR DE LA VENTANA Y FINALMENTE DONE UPLOADING.



```

void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH); // set the LED on
    delay(1000);           // wait for a second
    digitalWrite(13, LOW); // set the LED off
    delay(1000);           // wait for a second
}

```

UN SKETCH, COMO UN PROGRAMA ESCRITO EN CUALQUIER LENGUAJE, ES UN CONJUNTO DE INSTRUCCIONES PARA EL ORDENADOR. SI MIRAMOS CON DETENIMIENTO EL SKETCH BLINK VEREMOS QUE TIENE 2 PARTES PRINCIPALES, EL **SETUP** Y EL **LOOP**.

[HTTP://ARDUINO.CC/ES/REFERENCE/HOMEPAGE](http://arduino.cc/es/reference/homepage)



MIRA LA PAGINA WEB DE ARDUINO PARA CONSULTAR LA GUIA DE REFERENCIA Y OTROS RECURSOS PARA APRENDER EL LENGUAJE.

**SETUP:** SE EJECUTA UNA VEZ CUANDO EL PROGRAMA ARRANCA

**LOOP:** SE REPITE UNA Y OTRA VEZ

ESTOS BLOQUES DE CODIGO SE LLAMAN FUNCIONES Y CADA SKETCH DEBE TENERLOS. TODO SU CONTENIDO ESTA ENCERRADO ENTRE LLAVES {}.

```

void setup() {           //DECLARA UN BLOQUE DE CODIGO
    pinMode(13, OUTPUT); //PONE EL PIN 13 COMO SALIDA
}                         //FIN DEL BLOQUE DE CODIGO

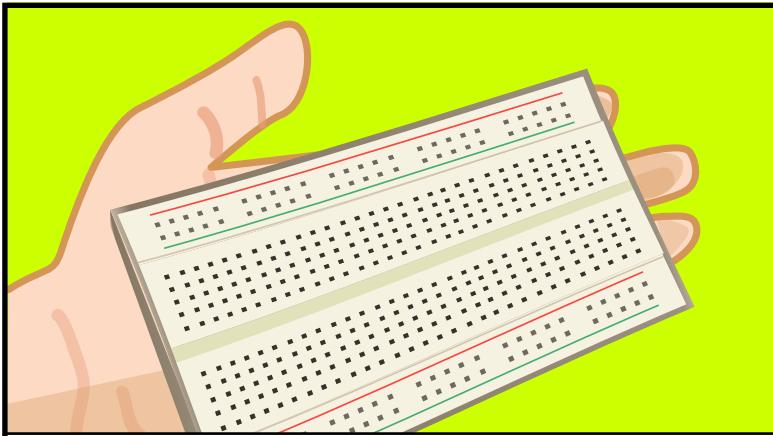
```

```

void loop() {             //DECLARA UN BLOQUE DE CODIGO
    digitalWrite(13, HIGH); //PONE EL PIN 13 EN ALTO
    delay(1000);           //PAUSA DE 1 SEGUNDO
    digitalWrite(13, LOW);  //PONE EL PIN 13 EN BAJO
    delay(1000);           //PAUSA DE 1 SEGUNDO
}                         //FIN DEL BLOQUE DE CODIGO

```

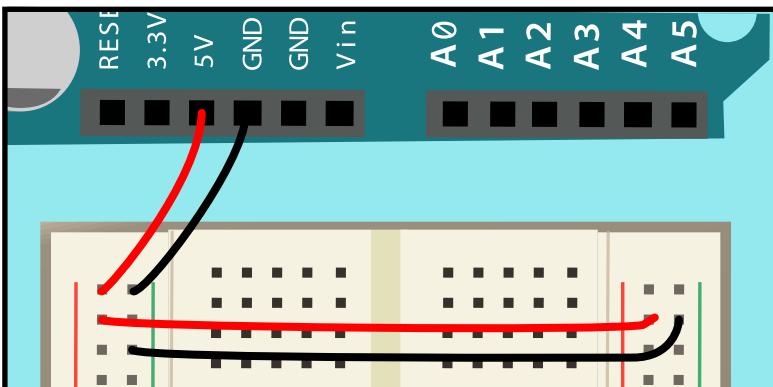
AHORA VAMOS A VER UN PROGRAMA SENCILLO LINEA POR LINEA Y VAMOS A VER QUE HACE CADA UNA.



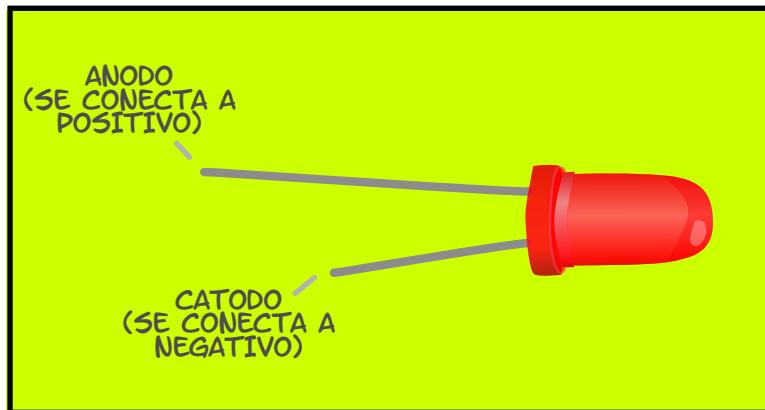
¿COMO PODEMOS CONTROLAR LOS OBJETOS QUE NO ESTAN EN LA PLACA ARDUINO? CONECTANDO LA ARDUINO A UNA PLACA DE PROTOTIPIADO SIN SOLDADURAS. ESTA NOS PERMITE MONTAR Y PROBAR CIRCUITOS RAPIDAMENTE.



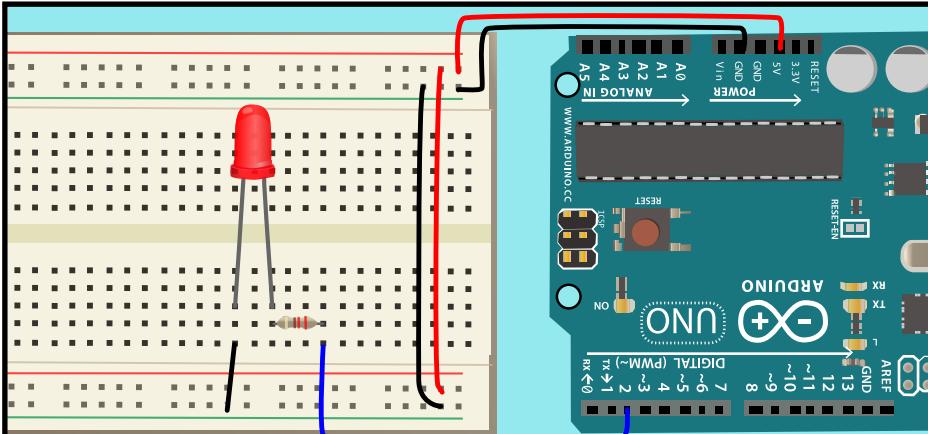
ESTA PLACA DE PROTOTIPIADO TIENE 2 FILAS DE AGUJEROS EN LOS BORDES DERECHO E IZQUIERDO Y 5 FILAS DE AGUJEROS A CADA LADO DE LA SEPARACION CENTRAL. LAS FILAS LATERALES ESTAN CONECTADAS VERTICALMENTE, CADA UNA DE LAS FILAS DE 5 AGUJEROS ESTAN CONECTADAS HORIZONTALMENTE.



CONECTAREMOS POSITIVO Y NEGATIVO DESDE ARDUINO A LAS TIRAS VERTICALES USANDO CABLES. LOS COMPONENTES SE PUEDEN ENCAJAR DESDE LOS AGUJEROS DEL CENTRO A POSITIVO O NEGATIVO SEGUN SE NECESITE.



CUANDO LA CORRIENTE FLUYE A TRAVES DE UN LED (DIOXIO EMISOR DE LUZ) EN EL SENTIDO CORRECTO SE ENCIENDE. CONECTAREMOS UN LED A LA PLACA DE PROTOTIPIADO, LUEGO AL ARDUINO Y LO CONTROLAREMOS CON CODIGO.



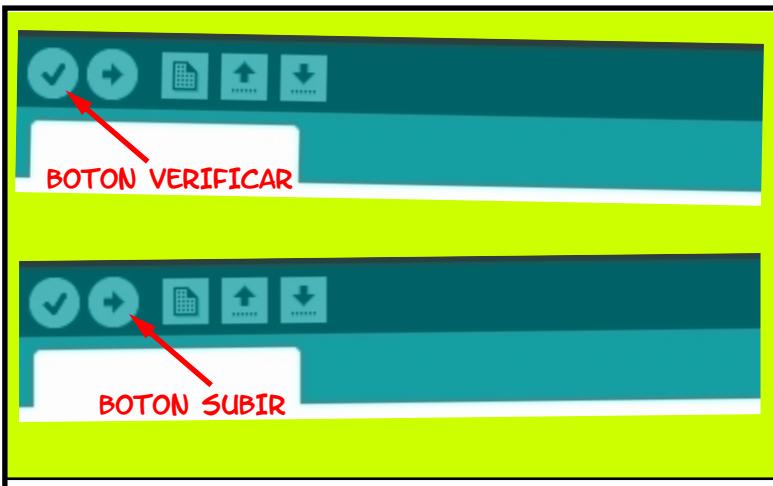
EL ANODO ESTA CONECTADO AL PIN 2 DE ARDUINO A TRAVES DE UNA RESISTENCIA DE 220 OHM. EL CATODO ESTA CONECTADO A NEGATIVO. LOS PINES DEL 2 AL 13 PUEDEN SER CONFIGURADOS COMO ENTRADAS O SALIDAS. PINCHA EN EL BOTON NEW PARA ABRIR UN NUEVO SKETCH.

```
void setup() {
    pinMode(2, OUTPUT);
}

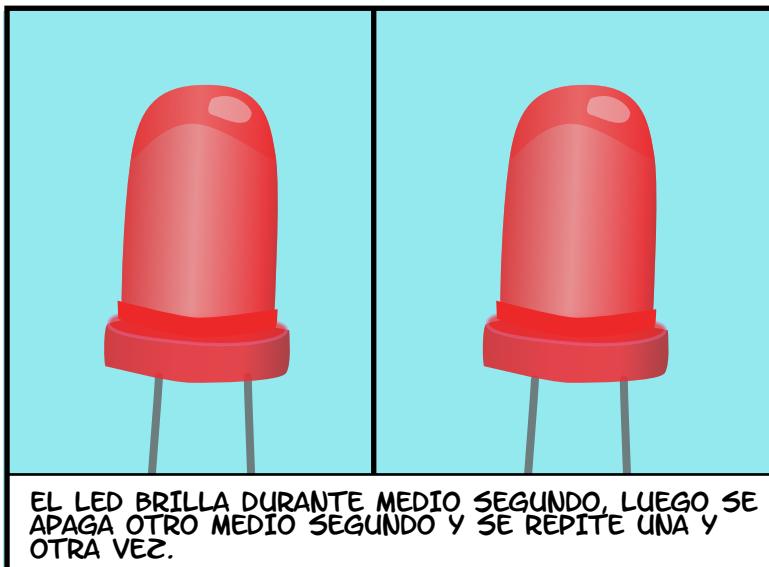
void loop() {
    digitalWrite(2, HIGH);
    delay(500);
    digitalWrite(2, LOW);
    delay(500);
}
```

EN EL SETUP PONEMOS EL PIN 2 COMO SALIDA.

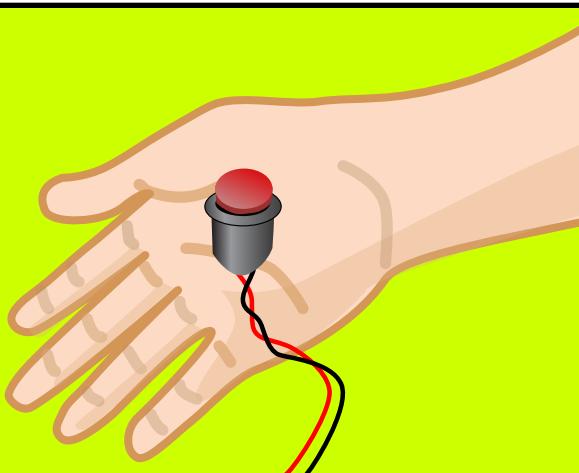
EN EL LOOP PRIMERO PONEMOS EL PIN 2 EN ESTADO ALTO PARA QUE EL LED SE ENCIENDA. EL DELAY HACE UNA PAUSA DE 500 MILISEGUNDOS, O MEDIO SEGUNDO. DESPUES SE PONE EL PIN 2 EN ESTADO BAJO Y EL LED SE APAGA. HACEMOS OTRA PAUSA DE MEDIO SEGUNDO.



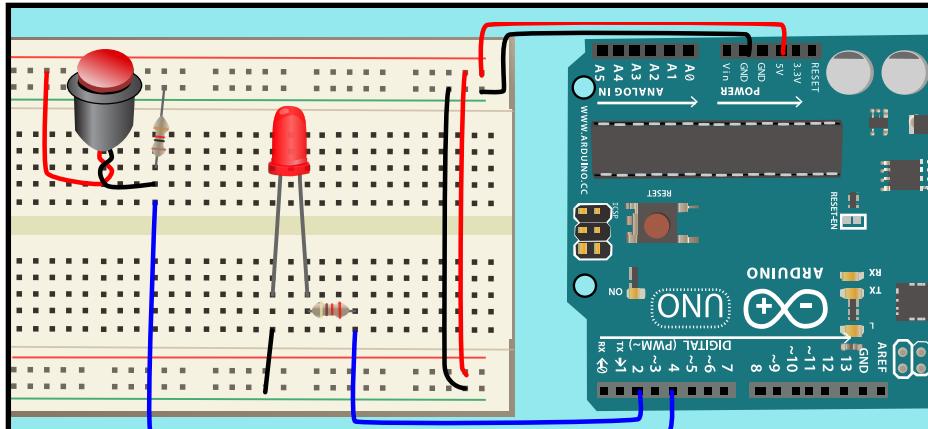
CLICAR EN VERIFICAR PARA COMPROBAR EL CODIGO. SI NO HAY ERRORES CLICAR EN SUBIR PARA PASAR EL PROGRAMA A LA ARDUINO.



EL LED BRILLA DURANTE MEDIO SEGUNDO, LUEGO SE APAGA OTRO MEDIO SEGUNDO Y SE REPITE UNA Y OTRA VEZ.



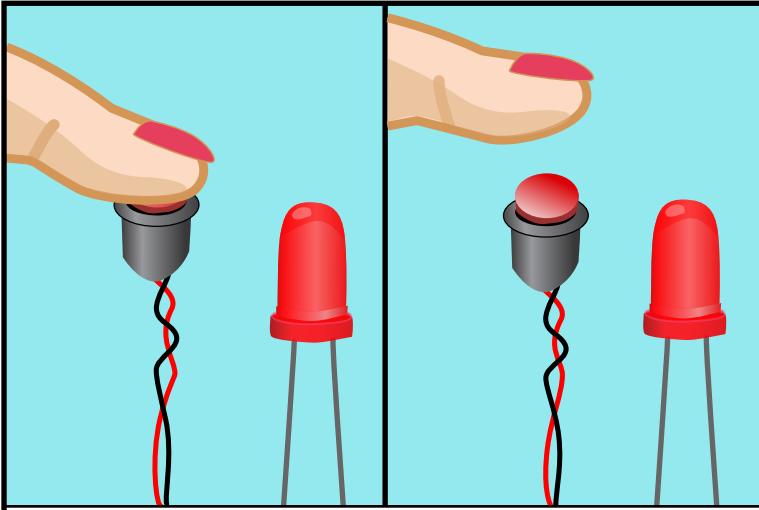
AHORA AÑADIREMOS UN PULSADOR, UNA ENTRADA DIGITAL, PARA ENCENDER Y APAGAR EL LED.



CONECTAR UN CABLE DEL PULSADOR AL PIN 4 DE ARDUINO, CON UNA RESISTENCIA DE 10K CONECTADA A NEGATIVO Y AL MISMO CABLE. CONECTAR EL OTRO CABLE A POSITIVO. DEJAMOS EL LED CONECTADO COMO ESTABA.

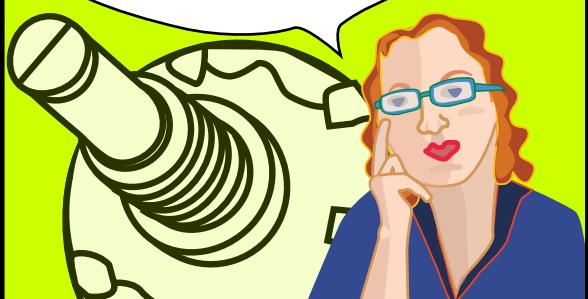
```
void setup() {  
    pinMode(2, OUTPUT);  
    pinMode(4, INPUT);  
}  
  
void loop() {  
    if(digitalRead(4)){  
        digitalWrite(2, HIGH);  
    }else{  
        digitalWrite(2, LOW);  
    }  
}
```

AHORA ESCRIBIMOS EL CODIGO. EN EL SETUP DECLARAMOS EL PIN 2 COMO SALIDA Y EL PIN 4 COMO ENTRADA. EN EL LOOP USAMOS UNA CONDICIÓN IF, SI LEE EL PIN 4 COMO ALTO PONE EL LED EN ALTO, DE LA MISMA FORMA LO LEE COMO BAJO EL LED SE APAGARA.

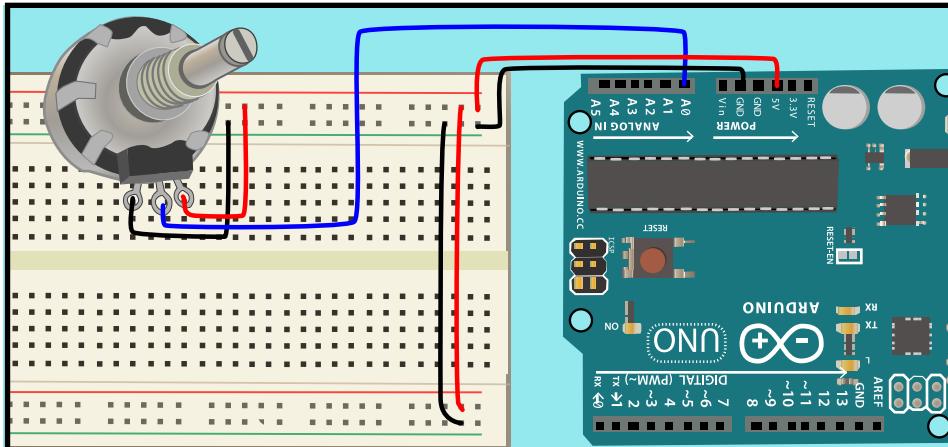


EL LED SE ENCIENDE CUANDO EL PULSADOR SE ACCIONA.

UN POTENCIOMETRO ES UNA RESISTENCIA VARIABLE. LA RESISTENCIA CAMBIA SEGUN SE GIRA SUBIENDO O BAJANDO DEPENDIENDO DE EN QUE SENTIDO LO GIRAMOS



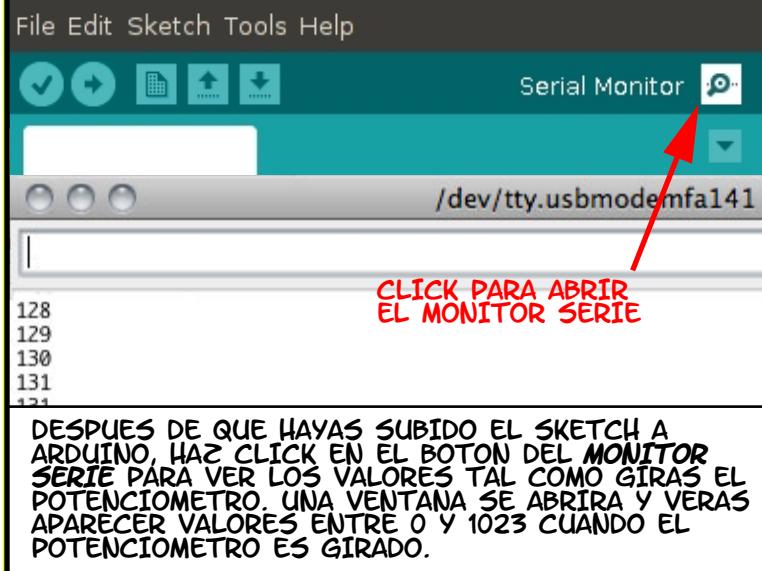
AHORA UTILIZAREMOS UNA ENTRADA ANALOGICA. USAREMOS UN POTENCIOMETRO.



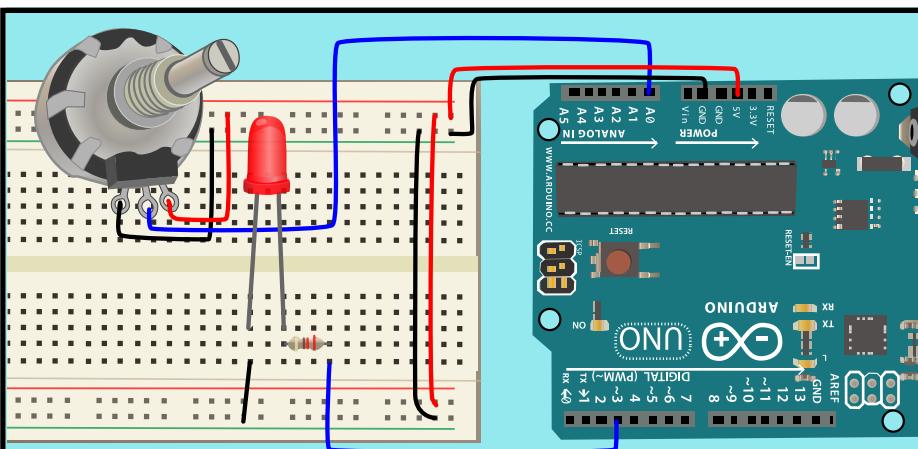
CONECTA EL PIN CENTRAL DEL POTENCIOMETRO AL PIN ANALOGICO A0. CONECTA UN EXTREMO DEL POTENCIOMETRO A POSITIVO Y EL OTRO A NEGATIVO.

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println(analogRead(A0));  
}
```

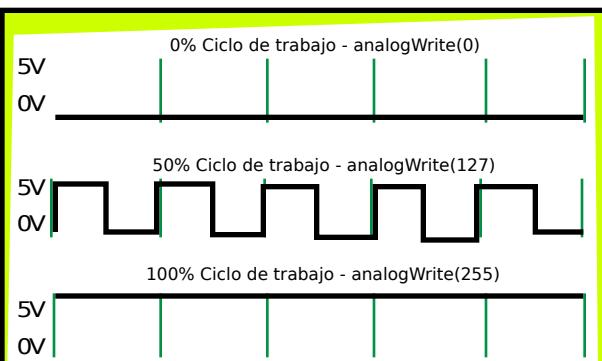
PRIMERO VEREMOS EL RANGO DE VALORES QUE TOMAMOS GIRANDO EL POTENCIOMETRO UTILIZANDO EL MONITOR SERIE. EN NUESTRO CODIGO INICIALIZAMOS EL OBJETO SERIAL EN EL SETUP, ESTABLECIENDO LA VELOCIDAD A 9600 BAUDIOS. EN EL LOOP LEEMOS EL VALOR DEL PIN ANALOGICO A0 Y LO ESCRIBIMOS EN EL OBJETO SERIAL UTILIZANDO LA FUNCION PRINTLN.



DESPUES DE QUE HAYAS SUBIDO EL SKETCH A ARDUINO, HAZ CLICK EN EL BOTON DEL MONITOR SERIE PARA VER LOS VALORES TAL COMO GIRAS EL POTENCIOMETRO. UNA VENTANA SE ABRIRA Y VERAS APARECER VALORES ENTRE 0 Y 1023 CUANDO EL POTENCIOMETRO ES GIRADO.



VAMOS A UTILIZAR LOS VALORES CAMBIANTES QUE RECIBIMOS DEL POTENCIOMETRO PARA CONTROLAR UN LED. CONECTA EL ANODO A TRAVES DE UNA RESISTENCIA AL PIN 3 Y EL CATODO A NEGATIVO.

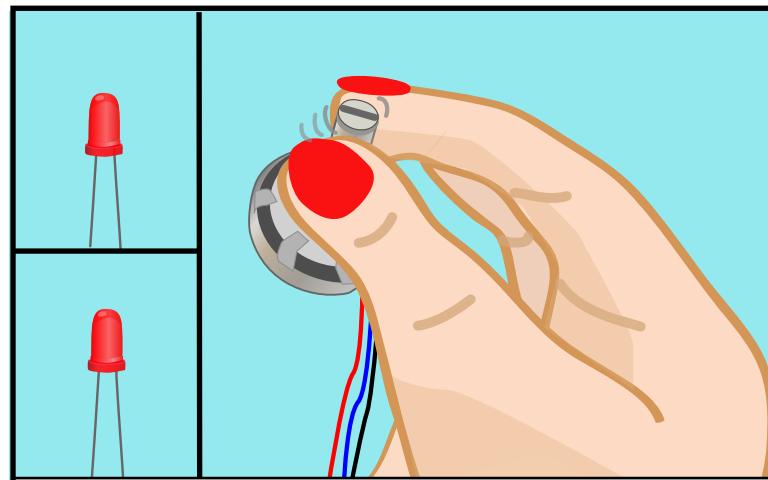


VAMOS A UTILIZAR EL PULSO CON MODULACION DE PROFUNDIDAD (PWM). ESTE ES UN METODO PARA SIMULAR UN VALOR ANALOGICO MANIPULANDO EL VOLTAJE, ACTIVANDOLO O DESACTIVANDOLO A INTERVALOS DIFERENTES, O CICLOS DE TRABAJO. PUEDES UTILIZAR EL PWM EN LOS PINES 3, 5, 6, 9, 10 Y 11.

```
int sensorValue = 0;
void setup() {
    pinMode(3, OUTPUT);
}
void loop() {
    sensorValue = analogRead(A0);
    analogWrite(3, sensorValue/4);
}
```

PRIMERO CREAMOS UNA VARIABLE PARA GUARDAR EN ELLA EL VALOR DEL POTENCIOMETRO. EN EL SETUP ESTABLECEMOS EL PIN 3 COMO SALIDA. EN EL LOOP GUARDAMOS EL VALOR QUE HEMOS LEIDO DEL PIN A0 EN NUESTRA VARIABLE.

LUEGO ESCRIBIMOS EL VALOR EN EL PIN 3, NUESTRO PIN PARA EL LED. DEBEMOS DIVIDIR EL VALOR DE LA VARIABLE POR 4, POR QUE MANEJAMOS VALORES ENTRE 0 Y 255, O UN BYTE.



EL BRILLO DEL LED CAMBIA, VARIANDO DE COMPLETAMENTE APAGADO A MUY BRILLANTE SEGUN GIRAMOS EL POTENCIOMETRO.



YA ESTA! ESTA ES UNA MUY BREVE INTRODUCCION. EN LAS PROXIMAS VINETAS ENCONTRAREIS LINKS Y OTROS RECURSOS PARA SEGUIR APRENDIENDO. PROVADLOS TODOS Y ENCONTRAREIS MUCHOS MAS.

## ENLACES SOFTWARE

### DESCARGAS

[HTTP://WWW.ARDUINO.CC/ES/MAIN/SOFTWARE](http://www.arduino.cc/es/main/software)

### REFERENCIAS DEL LENGUAJE

[HTTP://ARDUINO.CC/ES/REFERENCE/HOMEPAGE](http://arduino.cc/es/reference/homepage)

## DONDE COMPAR

### ARDUINO STORE

[HTTP://STORE.ARDUINO.CC/WW](http://store.arduino.cc/ww)

### ARDUMANIA.ES

[HTTP://WWW.ARDMANIA.ES/ARDUTIENDA](http://www.ardumania.es/ardutienda)

## TUTORIALES

### TUTORIALES EN ARDUINO .CC

[HTTP://WWW.ARDUINO.CC/ES/TUTORIAL/HOMEPAGE](http://www.arduino.cc/es/tutorial/homepage)

### TUTORIALES EN ARDUMANIA.ES

[HTTP://WWW.ARDMANIA.ES SECCION APRENDE](http://www.ardumania.es/seccion-aprende)

## ELECTRONICA

### TXAPUZAS.BLOGSPOT

[HTTP://TXAPUZAS.BLOGSPOT.COM](http://txapuzas.blogspot.com)

## PDFS EN CASTELLANO

### ARDUINO.CC/PLAYGROUND/ES

[HTTP://ARDUINO.CC/PLAYGROUND/ES/MANUALES](http://arduino.cc/playground/es/manuales)

### ARDUMANIA.ES

[HTTP://WWW.ARDMANIA.ES/DESCARGAS](http://www.ardumania.es/descargas)

TEXTOS Y DIBUJOS POR **JODY CULKIN** PARA VER MAS [JODYCULKIN.COM](http://jodyculkin.com)

TRADUCCION AL CASTELLANO POR **JOSE MANUEL ESCUDER** ([ARDUMANIA.ES](http://ardumania.es))

AGRADECIMIENTOS A A TOM IGUE, MARIANNE PETIT, CALVIN REID, LA FACULTAD Y STAFF DEL INTERACTIVE TELECOMMUNICATIONS PROGRAM EN NY, PARTICULARMENTE A DAN O'SULLIVAN, DANNY ROZIN Y RED BURNS. GRACIAS A CINDY KARASEK, CHRIS STEIN, SARAH TEITLER, KATHY GONCHAROV & ZANNAH MARSH.

MUCHAS MUCHAS GRACIAS PARA EL ARDUINO TEAM POR OFRECERNOS ESTA ROBUSTA Y FLEXIBLE PLATAFORMA OPEN SOURCE.

Y GRACIAS A LA VIVA, ACTIVA Y CRECIENTE COMUNIDAD DE ARDUINO, EN ESPECIAL A LA DE HABLA HISPANA.

INTRODUCCION A ARDUINO POR JODY CULKIN ESTA BAJO LICENCIA CREATIVE COMMONS ATRIBUCION - NO COMERCIAL - COMPARTIR IGUAL 3.0