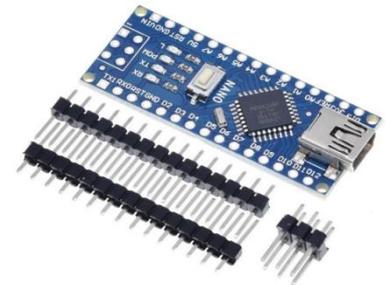
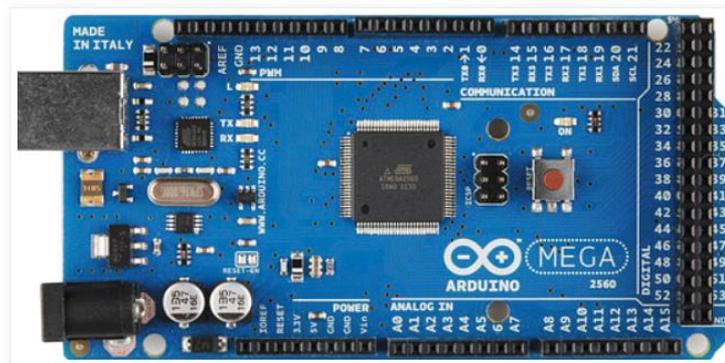


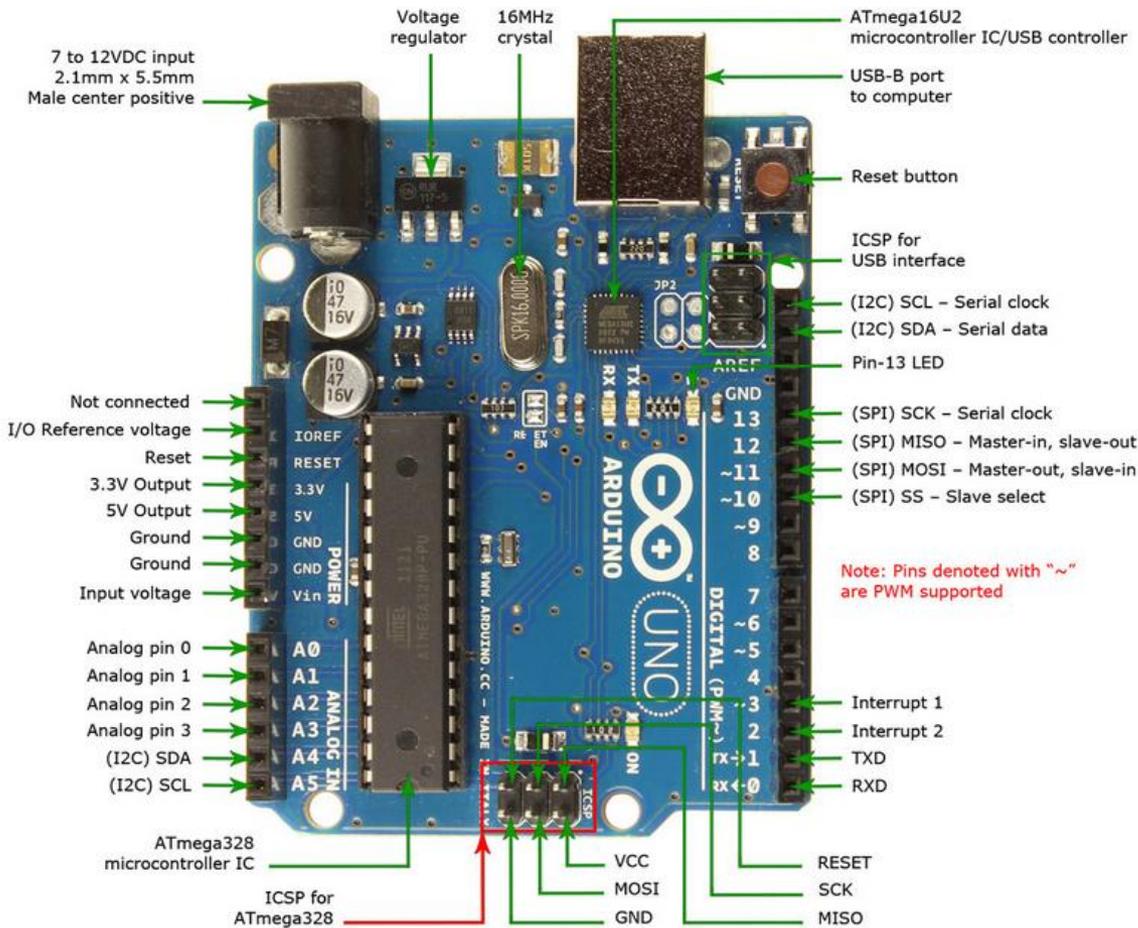
ARDUINO

QUE ES ARDUINO

- El **arduino** (nace en 2005 en Italia, por estudiantes) es una placa que tiene todos los elementos necesarios **para** conectar periféricos a las entradas y salidas de un microcontrolador. Es decir, es una placa impresa con los componentes necesarios **para** que funcione el microcontrolador y su comunicación con un ordenador a través de la comunicación serial.



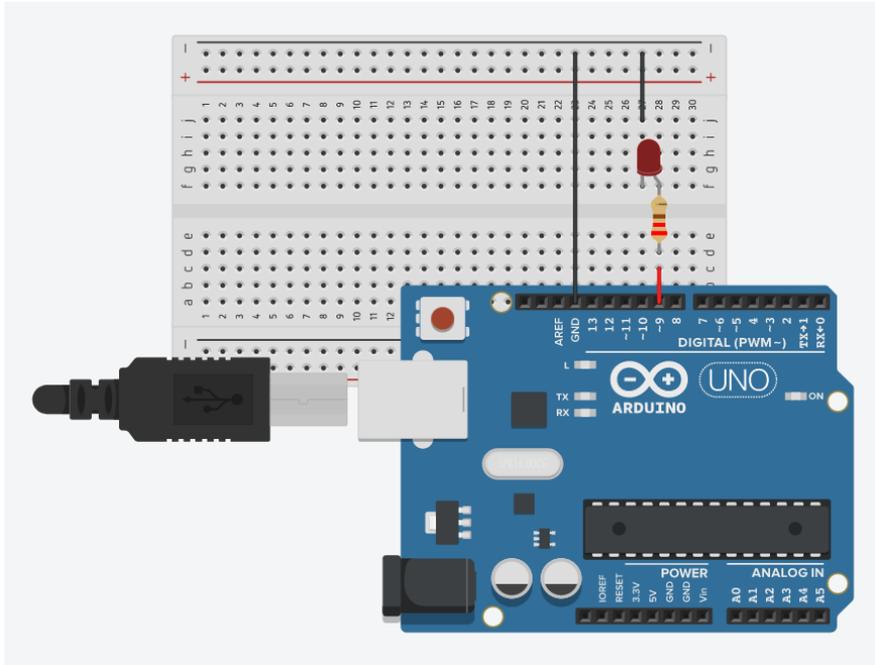
Partes de la placa



Technical Details:

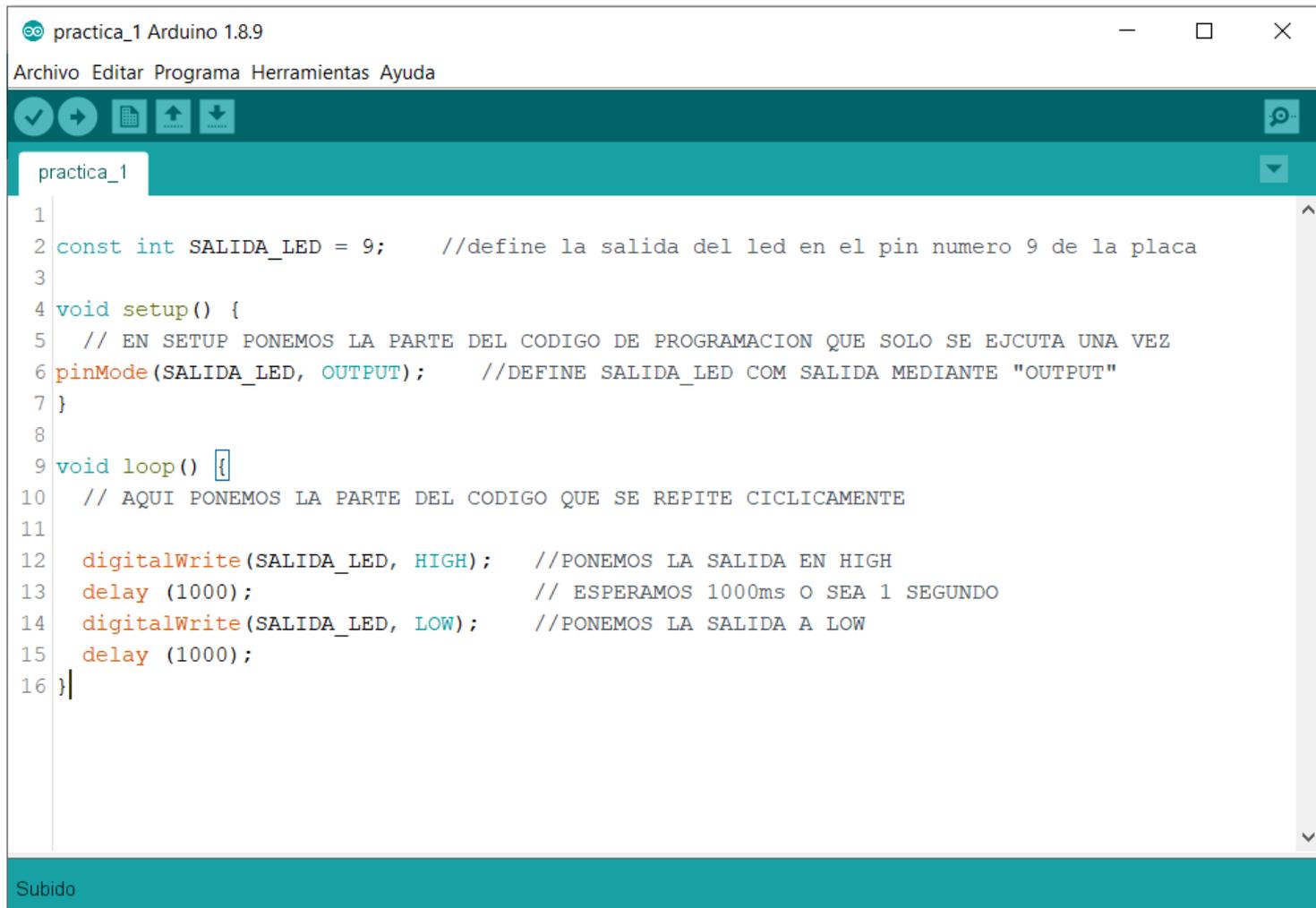
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Practica 1, SALIDAS DIGITALES



Conectamos el circuito según el esquema: en las resistencias de nuestro kit escogemos una de colores:
rojo-rojo-negro-negro-tolerancia

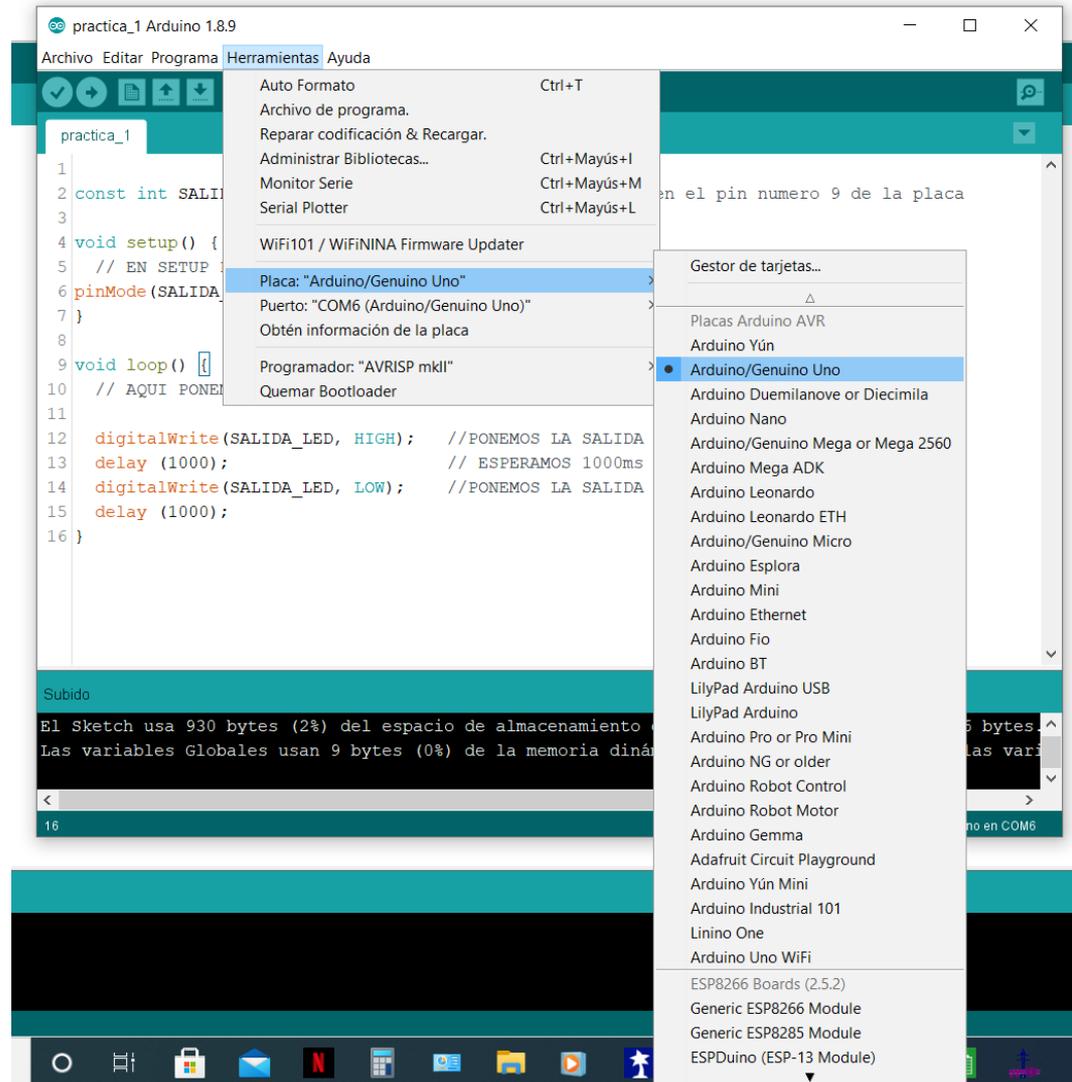
El programa



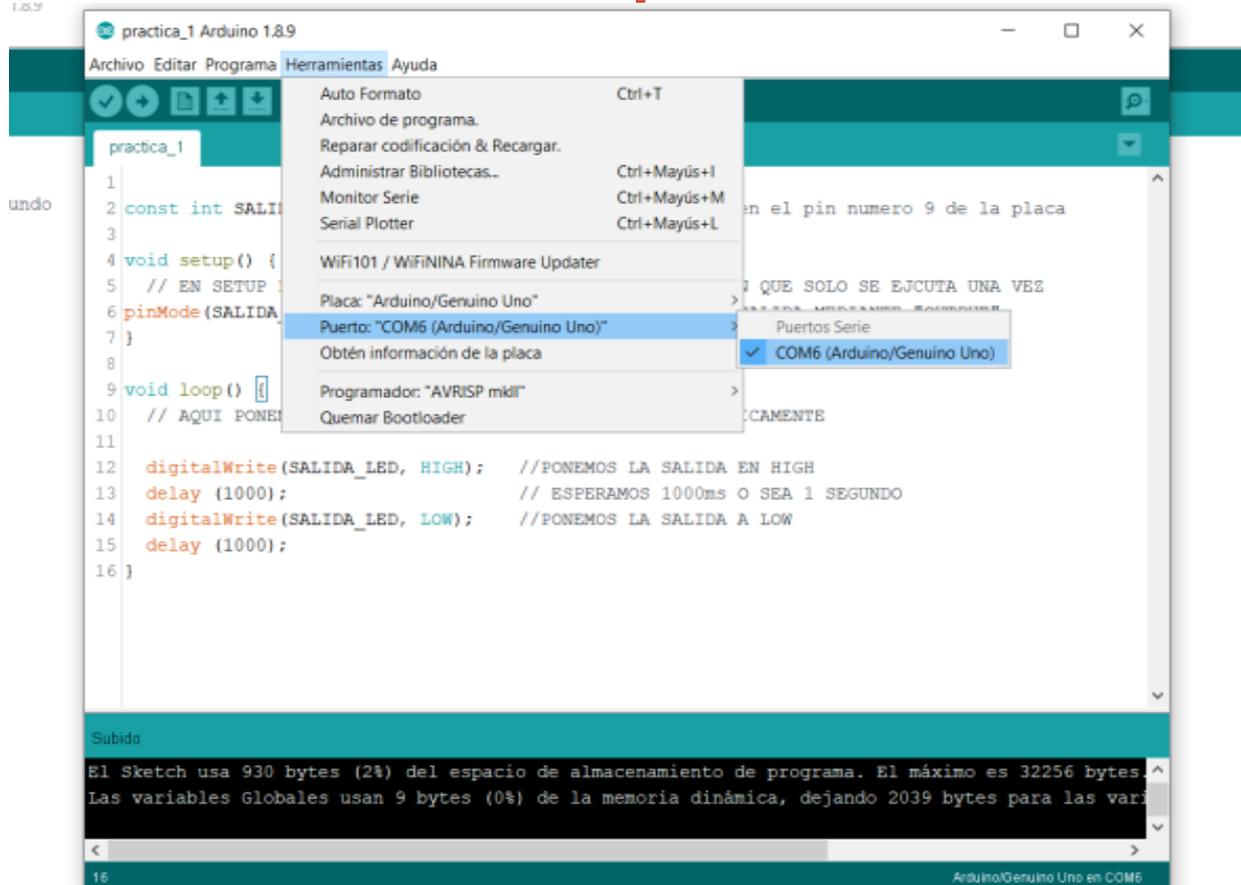
```
practica_1 Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
practica_1
1
2 const int SALIDA_LED = 9;    //define la salida del led en el pin numero 9 de la placa
3
4 void setup() {
5     // EN SETUP PONEMOS LA PARTE DEL CODIGO DE PROGRAMACION QUE SOLO SE EJCUTA UNA VEZ
6     pinMode(SALIDA_LED, OUTPUT);    //DEFINE SALIDA_LED COM SALIDA MEDIANTE "OUTPUT"
7 }
8
9 void loop() {
10    // AQUI PONEMOS LA PARTE DEL CODIGO QUE SE REPITE CICLICAMENTE
11
12    digitalWrite(SALIDA_LED, HIGH);    //PONEMOS LA SALIDA EN HIGH
13    delay (1000);    // ESPERAMOS 1000ms O SEA 1 SEGUNDO
14    digitalWrite(SALIDA_LED, LOW);    //PONEMOS LA SALIDA A LOW
15    delay (1000);
16 }
```

Subido

Seleccionar el tipo de Arduino



Seleccionar el puerto de salida USB



Compilar y enviar al microcontrolador

- Se puede verificar (compilar) si el programa está bien escrito, o compilar y enviar directamente, si no tiene errores.



The screenshot shows the Arduino IDE interface. At the top, there is a menu bar with 'Archivo', 'Editar Programa', 'Herramientas', and 'Ayuda'. Below the menu bar is a toolbar with several icons, including a green checkmark and a green arrow, which are circled in red. The main area is a code editor with a dark background and light text. The code is as follows:

```
1
2 const int SALIDA_LED = 9;    //define la salida del led en el pin numero 9 de la placa
3
4 void setup() {
5   // EN SETUP PONEMOS LA PARTE DEL CODIGO DE PROGRAMACION QUE SOLO SE EJECUTA UNA VEZ
6   pinMode(SALIDA_LED, OUTPUT); //DEFINE SALIDA_LED COM SALIDA MEDIANTE "OUTPUT"
7 }
8
9 void loop() {
10  // AQUI PONEMOS LA PARTE DEL CODIGO QUE SE REPITE CICLICAMENTE
11
12  digitalWrite(SALIDA_LED, HIGH); //PONEMOS LA SALIDA EN HIGH
13  delay (1000); // ESPERAMOS 1000ms O SEA 1 SEGUNDO
14  digitalWrite(SALIDA_LED, LOW); //PONEMOS LA SALIDA A LOW
15  delay (1000);
16 }
```

At the bottom of the IDE, there is a status bar with the text 'Subido' and a message: 'El Sketch usa 930 bytes (2%) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las vari'. The page number '16' is visible in the bottom left corner, and 'Arduino/Genuino Uno en COM6' is visible in the bottom right corner.

Código de colores para resistencias: con 4 y 5 bandas

Código de colores para resistencias con 4 bandas



COLOR	BANDA 1	BANDA 2	MULTIPLICADOR	TOLERANCIA
NEGRO	0	0	x 1Ω	
MARRON	1	1	x 10Ω	±1%
ROJO	2	2	x 100Ω	±2%
NARANJA	3	3	x 1KΩ	
AMARILLO	4	4	x 10KΩ	
VERDE	5	5	x 100KΩ	
AZUL	6	6	x 1MΩ	
VIOLETA	7	7		
GRIS	8	8		
BLANCO	9	9		
DORADO			x 0,1Ω	±5%
PLATEADO			x 0,01Ω	±10%
		SIN BANDA		±20%

WWW.INVENTABLE.EU

Código de colores para resistencias de precisión con 5 bandas

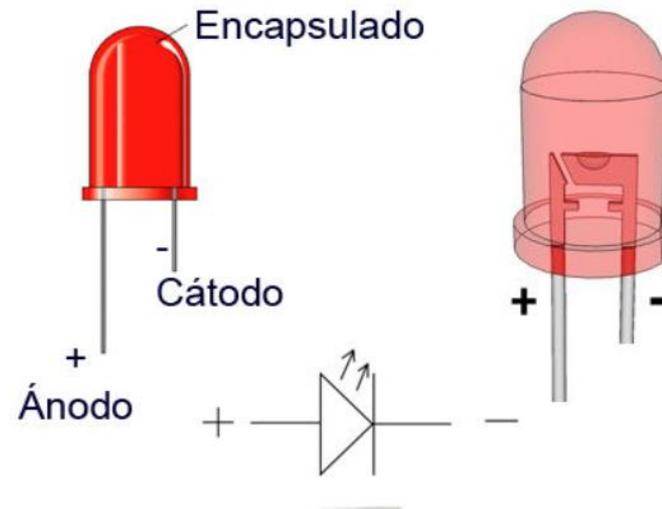
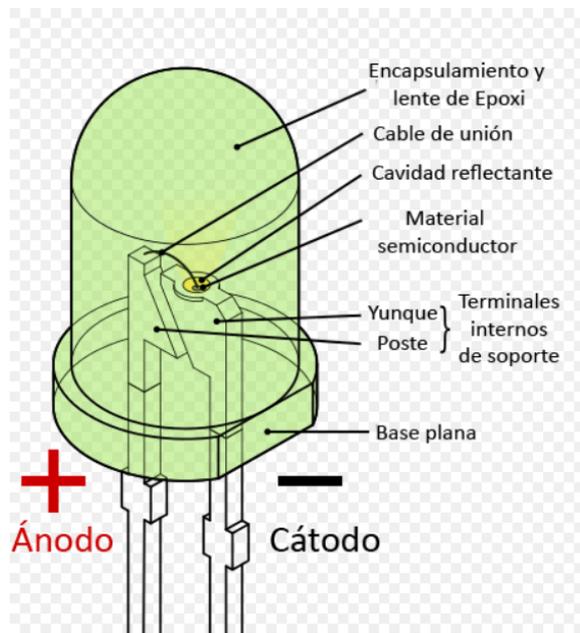


COLOR	BANDA 1	BANDA 2	BANDA 3	MULTIPLICADOR	TOLERANCIA
NEGRO	0	0	0	x 1Ω	
MARRON	1	1	1	x 10Ω	±1%
ROJO	2	2	2	x 100Ω	±2%
NARANJA	3	3	3	x 1KΩ	
AMARILLO	4	4	4	x 10KΩ	
VERDE	5	5	5	x 100KΩ	±0,5%
AZUL	6	6	6	x 1MΩ	±0,25%
VIOLETA	7	7	7	x 10MΩ	±0,10%
GRIS	8	8	8		±0,05%
BLANCO	9	9	9		
DORADO				x 0,1Ω	
PLATEADO				x 0,01Ω	

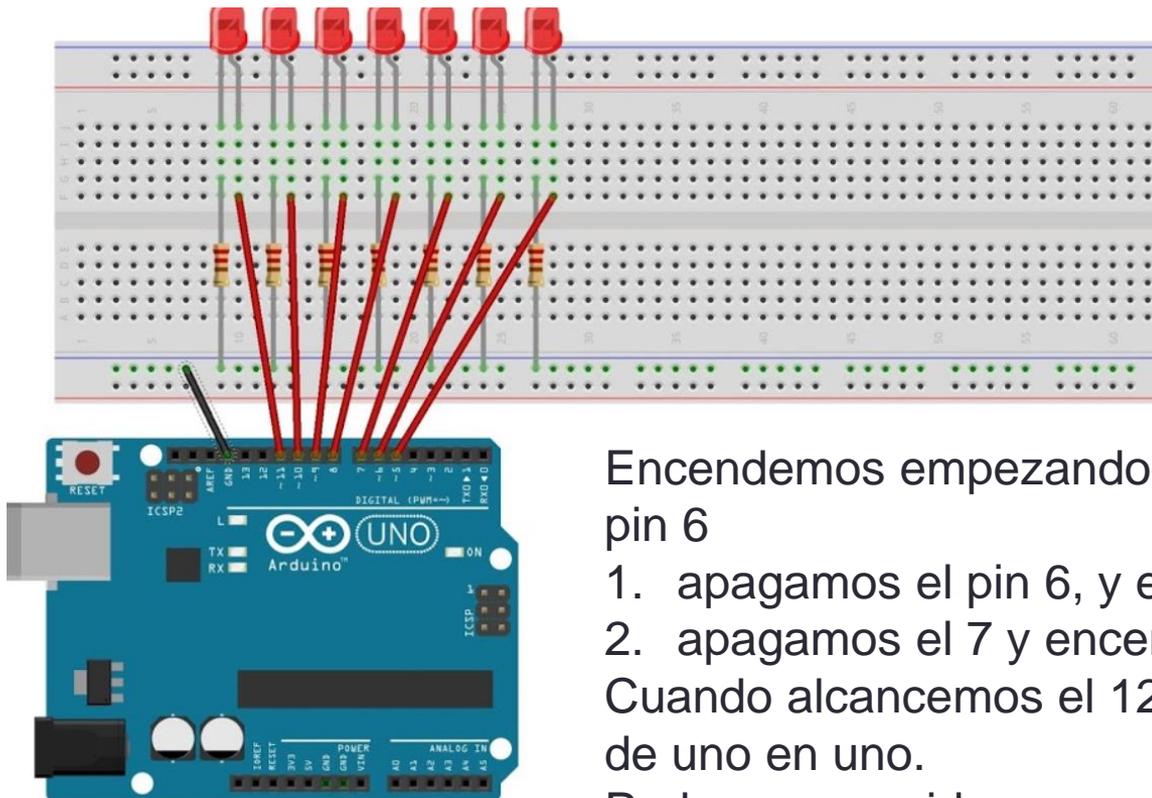
WWW.INVENTABLE.EU

Los leds

Un led es un diodo emisor de luz, formado por la unión de dos materiales semiconductores, unidos a los dos terminales: Ánodo (positivo) y el Cátodo (negativo).



Practica 2: juego de 7 leds



Encendemos empezando por el led conectado al pin 6

1. apagamos el pin 6, y encendemos el pin 7,
 2. apagamos el 7 y encendemos el pin 8,
- Cuando alcancemos el 12, apagamos, y volvemos de uno en uno.

Podemos ver video en-enlace inferior.

<https://youtu.be/-7Qa6zyP3Cg>

Practica 2: juego de 7 leds

```
1 const int led1=12;
2 const int led2=11;
3 const int led3=10;
4 const int led4=9;
5 const int led5=8;
6 const int led6=7;
7 const int led7=6;
8
9 int tiempo=100; //define una variable que utilizaremos en todos los delays
10
11 // defino conexion lampara
12
13 void setup() {
14     pinMode (led1,OUTPUT);
15     pinMode (led2,OUTPUT);
16     pinMode (led3,OUTPUT);
17     pinMode (led4,OUTPUT);
18     pinMode (led5,OUTPUT);
19     pinMode (led6,OUTPUT);
20     pinMode (led7,OUTPUT);
21     }
22
```

Practica 2: juego de 7 leds

```
23 void loop() {
24   digitalWrite(led1, HIGH);
25   digitalWrite(led2, LOW);
26   delay(tiempo);
27   digitalWrite(led1, LOW);
28   digitalWrite(led2, HIGH);
29   delay(tiempo);
30   digitalWrite(led2, LOW);
31   digitalWrite(led3, HIGH);
32   delay(tiempo);
33   digitalWrite(led3, LOW);
34   digitalWrite(led4, HIGH);
35   delay(tiempo);
36   digitalWrite(led4, LOW);
37   digitalWrite(led5, HIGH);
38   delay(tiempo);
39   digitalWrite(led5, LOW);
40   digitalWrite(led6, HIGH);
41   delay(tiempo);
42   digitalWrite(led6, LOW);
43   digitalWrite(led7, HIGH);
44   delay(tiempo);           //fin hacia un lado
```

Practica 2: juego de 7 leds

```
46  digitalWrite(led7, LOW);
47  digitalWrite(led6, HIGH);
48  delay(tiempo);
49  digitalWrite(led6, LOW);
50  digitalWrite(led5, HIGH);
51  delay(tiempo);
52  digitalWrite(led5, LOW);
53  digitalWrite(led4, HIGH);
54  delay(tiempo);
55  digitalWrite(led4, LOW);
56  digitalWrite(led3, HIGH);
57  delay(tiempo);
58  digitalWrite(led3, LOW);
59  digitalWrite(led2, HIGH);
60  delay(tiempo);
61                                     //NO ponemos la linea de encender led1 y apagar led2 porque ya esta en el inicio del ciclo
62                                     //SINO el led1 estaria encendido el doble de tiempo
63
64 }
```

Practica 3: con el bucle “for” y matriz[]

for (inicialización;comprobacion;incremento++ ó decremento--)
{.....instrucciones.....}

The **initialization** happens first and exactly once. Each time through the loop, the **condition** is tested; if it's true, the statement block, and the **increment** is executed, then the **condition** is tested again. When the **condition** becomes false, the loop ends.

```
practica_3$
1 /*
2  * COCHE FANTASTICO
3  */
4
5 int pinLEDS[7]={7,8,9,10,11,12,13}; //definimos los pines de cada LED
6 int i=0; //definimos una variable "i" que toma de valor inicial 0
7
8 void setup() { //configuracion de Arduino
9   for(i=0;i<7;i++)
10  {
11   pinMode(pinLEDS[i],OUTPUT); //configuramos las salidas mediante una matriz
12  }
13 }
14
15 void loop() { //Programa
16   for(i=0;i<7;i++)
17   {
18    digitalWrite(pinLEDS[i],HIGH); //endecemos un led y lo apagamos,
19    delay(150); //sucesivamente hasta el ultimo
20    digitalWrite(pinLEDS[i-1],LOW); //podemos escribir [i-1] Aparenta más dinamismo
21    delay(150);
22   }
23
24   for(i=6;i>=0;i--)
25   {
26    digitalWrite(pinLEDS[i],HIGH); //encendemos el último LED y lo apagamos
27    delay(150); //y así sucesivamente
28    digitalWrite(pinLEDS[i+1],LOW); //Y aquí [i+1]
29    delay(150);
30   }
31 }
```

