

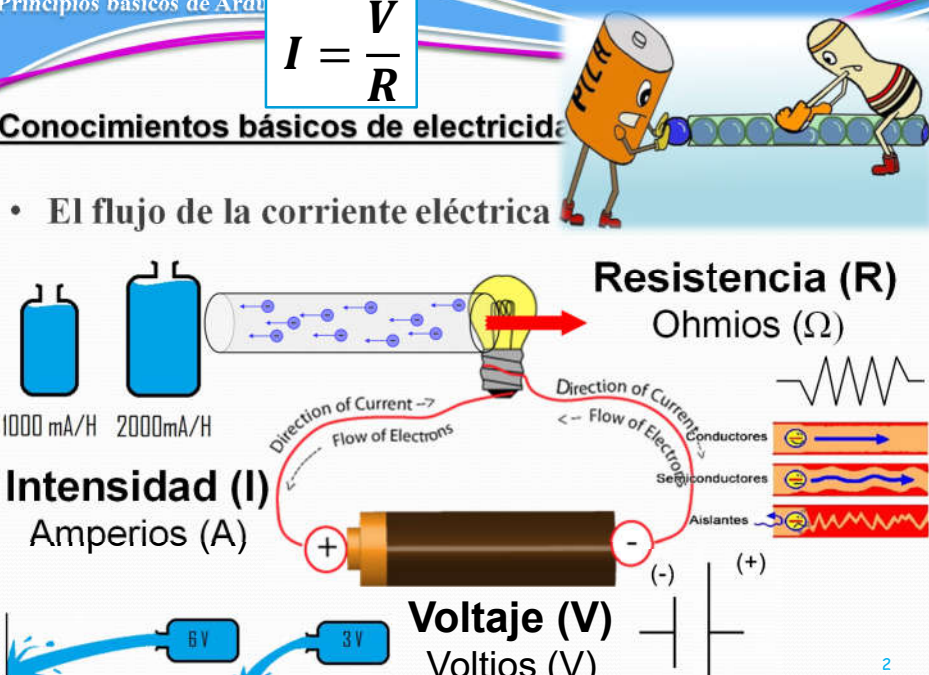


Principios básicos de Ardu

$$I = \frac{V}{R}$$

## Conocimientos básicos de electricidad

- El flujo de la corriente eléctrica



**Resistencia (R)**  
Ohmios ( $\Omega$ )

**Intensidad (I)**  
Amperios (A)

**Voltaje (V)**  
Voltios (V)

1000 mA/H 2000mA/H

Direction of Current  $\rightarrow$   
Flow of Electrons  $\leftarrow$

Conductores  
Semiconductores  
Aislantes

(-) (+)

2

Pr... de Arduino

$$I = \frac{V}{R}$$

### Conocimientos básicos de electricidad

- El flujo de la corriente eléctrica

Aplicación	Electricidad	Unidad
Motor (velocidad)	Potencial (Voltios)	Voltios (V) J/C
Cantidad de electrones	Intensidad ( $I = \frac{q}{t}$ )	Amperio (A) C/s
Impedimento al paso de corriente (TV, ordenadores, bombillas,...)	Resistencia (R)	Ohmio ( $\Omega$ )

3

3

### 1,5 V Intensidad o Voltaje

$I = 2,16 \text{ mA}$

**Circuito**

5000 volts?

**La pistola Taser**

**40.000 V**  
 $I = \text{¡¡}3 \text{ A!!!}$

$I = 0,21 \text{ mA}$

**CÓMO FUNCIONAN?**

50.000V

20 mA

$I = 0,005 \text{ A}$   
 $= 5,0 \text{ miliamperios (mA)}$   
Sólo paraliza

cables que transmiten de hasta 7,6 metros

la ropa de la víctima y efectúan segundos que contrae la

Umbral de percepción

Nivel de seguridad aceptable (5 mA)

Pérdida de control motor (12 mA)

Dolor Fatiga Lesiones físicas

Fibrilación ventricular

Contracción sostenida de músculo cardíaco

Quemaduras Lesiones varias

3

4

Resistencia cuerpo humano: 1000 Ω

$I = \frac{V}{1000 \Omega} \leq 20 \text{ mA}$

**Conocimientos básicos de electricidad**

- El flujo de la corriente eléctrica

Las pilas

**MN2400**  
Size: AAA  
Chemistry: Alkaline  
Nominal Voltage: 1.5  
Max. Diameter: .413"  
Max. Height: 1.752"  
Capacity (mAh): 1,150  
Primary Usage: Flashlights, Calculators, Pagers, Remote Controls



(Batteries are shown approximately actual size).

**MN1500**  
Size: AA  
Chemistry: Alkaline  
Nominal Voltage: 1.5  
Max. Diameter: .571"  
Max. Height: 1.988"  
Capacity (mAh): 2,850  
Primary Usage: Flashlights, Radios, Recorders, Toys, Calculators, Pagers, CD Players, Electronic Games



**MN1400**  
Size: C  
Chemistry: Alkaline  
Nominal Voltage: 1.5  
Max. Diameter: 1.031"  
Max. Height: 1.969"  
Capacity (mAh): 7,800  
Primary Usage: Flashlights, Radios, Recorders, Toys



**MN1300**  
Size: D  
Chemistry: Alkaline  
Nominal Voltage: 1.5  
Max. Diameter: 1.346"  
Max. Height: 2.421"  
Capacity (mAh): 15,000  
Primary Usage: Flashlights, Radios, Recorders, Toys



5

Resistencia cuerpo humano: 1000 Ω

$I = \frac{V}{1000} \leq 0,02 \text{ A}$

**Tercer paso: Elementos de un circuito**

**$V = I \cdot R = 0,02 \cdot 1000 \leq 20 \text{ V}$**

- El flujo de la corriente eléctrica

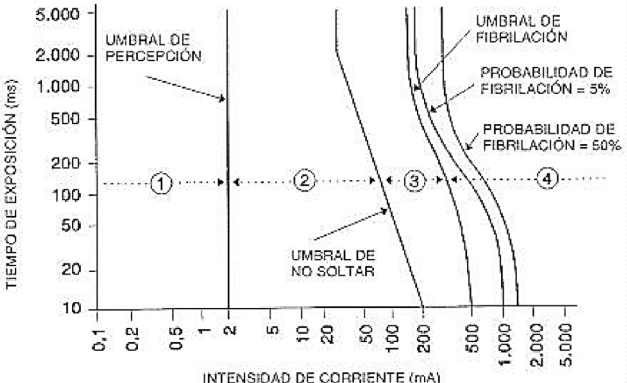
Corriente Continua

**Zona 1:** ninguna reacción.

**Zona 2:** ningún efecto fisiológico peligroso.

**Zona 3:** ningún daño orgánico. contracciones musculares, paradas temporales del corazón

**Zona 4:** riesgo de parada cardíaca, quemaduras graves,...



The graph plots 'TIEMPO DE EXPOSICIÓN (ms)' on a logarithmic y-axis (10 to 5,000) against 'INTENSIDAD DE CORRIENTE (mA)' on a logarithmic x-axis (0.1 to 5,000). It shows several curves representing different risk levels: 'UMBRAL DE PERCEPCIÓN' at approximately 0.5 mA, 'UMBRAL DE NO SOLTAR' at approximately 10 mA, 'UMBRAL DE FIBRILACIÓN' at approximately 100 mA, and two curves for 'PROBABILIDAD DE FIBRILACIÓN = 5%' and 'PROBABILIDAD DE FIBRILACIÓN = 50%'. Four zones are marked with circled numbers 1 through 4 along the x-axis.

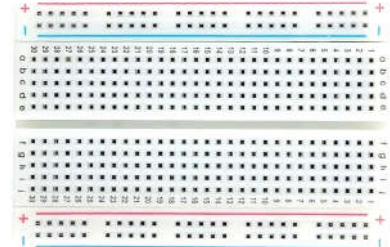


6

Principios básicos de Arduino

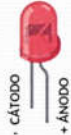
- Abre la caja

**Primer paso**

- Localiza la placa Arduino



- Sacar la placa Protoboard
- Sacar el cable USB
- Sacar un LED rojo


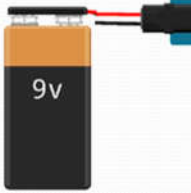
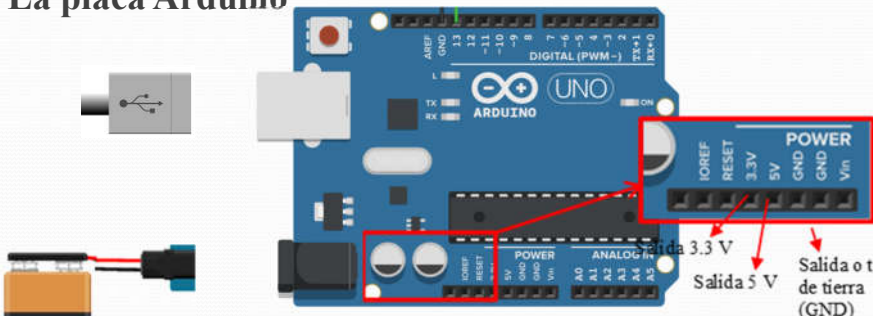


7

Principios básicos de Arduino

**Conocimientos básicos de electricidad**

- La placa Arduino



Salida 3.3 V

Salida 5 V

Salida o toma de tierra (GND)

GND (masa, tierra) 0 V

GrouND

5 V

8

Principios básicos de Arduino

### Conocimientos básicos

- La placa Protoboard

Los 5 agujeros de cada fila horizontal están conectados eléctricamente a través de las tiras de metal en el interior de la placa de pruebas

La fila del medio rompe la conexión entre los dos lados de la placa

Las tiras verticales que recorren toda la longitud de la placa está eléctricamente conectada. Estas tiras se suelen usar para las conexiones de alimentación y masa.

La parte superior de la placa de pruebas y las conexiones que hay debajo

BUS DE ALIMENTACIÓN

Área de montaje

BUS DE ALIMENTACIÓN

Salida 3.3 V

Salida 5 V

Salida o toma de tierra (GND)

9

Principios básicos de Arduino

### Conocimientos básicos

- La placa Protoboard

Salida 3.3 V

Salida 5 V

Salida o toma de tierra (GND)

50 mA

435 mA

Tiras metálicas conductoras

Las tiras conductoras en el interior de la placa de pruebas.

Figura 4

CÁTODO

ÁNODO

1,7 V / 20 mA

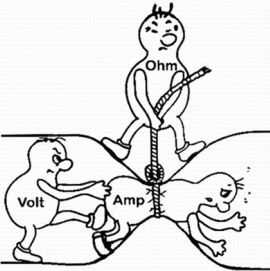
10

Principios básicos de Arduino

**435 mA** **1,7 V / 20 mA**

**Conocimientos básicos de electricidad**


- La ley de Ohm:


$$I = V / R$$


$$R = V / I$$


$$R = V / I$$

$V = I \cdot R$ 
 $I = V / R$ 
 $R = V / I$





**20 mA**  
**0,020 A**



**5 V**  
**250 Ω**

Corriente, voltaje y resistencia están todos relacionados. Cuando cambia uno de estos parámetros en un circuito, afecta a los demás. La relación que existe entre ellos se conoce como ley de Ohm, en honor a Georg Simon Ohm quien la descubrió.

**TENSIÓN (V) = CORRIENTE (I) x RESISTENCIA**

Al medir intensidad (amperios) en los circuitos que vaya a montar, los valores serán del rango de miliamperios. Un miliamperio vale la milésima parte de un amperio.

Ley de Ohm

11

11

Principios básicos de Arduino

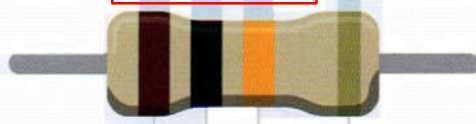
**Conocimientos básicos de electricidad**

- Primer circuito Tolerancia = error


4 BANDAS

1 0 x 10<sup>3</sup>


± 5 = 10.000Ω = 10kΩ ±5%




**220 Ω**



220Ω



560Ω



4.7kΩ

1 <sup>er</sup> DIGIT	2 <sup>er</sup> DIGIT
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

**Negro Marrón**

MULTIPLIER	TOLERANCE
0	+1%
1	+2%
2	+5% GOLD
3	+10% SILVER
4	
5	
6	

12

12

Principios básicos de Arduino

### Conocimientos básicos de electricidad

- Primer circuito

Las que vienen en el kit

The chart shows color bands for digits 0-9, multipliers, and tolerances. A resistor with bands 1, 0, 0, and a multiplier of 5 is shown, labeled as 10kΩ ± 5%. Three resistors from the kit are shown: 220 Ω (brown, red, black, gold), 560 Ω (blue, green, black, gold), and 4.7 K Ω (yellow, violet, black, orange, brown).

13

13

Principios básicos de Arduino

### Conocimientos básicos de electricidad

- Placa como suministro de voltaje

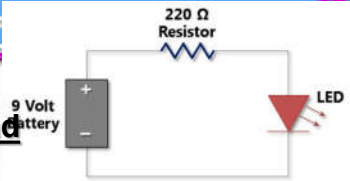
The diagram shows an Arduino board connected to a breadboard. A 9V battery is connected to the breadboard. A 220 Ω resistor is connected between the positive terminal of the battery and the anode of an LED. The cathode of the LED is connected to the GND pin of the Arduino board. The breadboard is labeled with pins 1 through 30.

14

14

Principios básicos de Arduino

**Conocimientos básicos de electricidad**



- Placa como suministro de voltaje

**Actividad 1.** Cambia la resistencia de  $220\ \Omega$  por otra de  $1\ K\Omega$ . Pero antes de realizarlo, ¿Qué crees que le pasará al brillo del LED? averigüemos tu idea previa:

- (a) El brillo del LED será mayor.
- (b) El brillo del LED será igual.
- (c) El brillo del LED será menor.

*¿Porqué?*

15

Principios básicos de Arduino

**Primer paso**

- Abre la caja
- Localiza la placa Arduino
- Abrir el programa de Arduino o instalar
- Conecta la placa al ordenador por USB

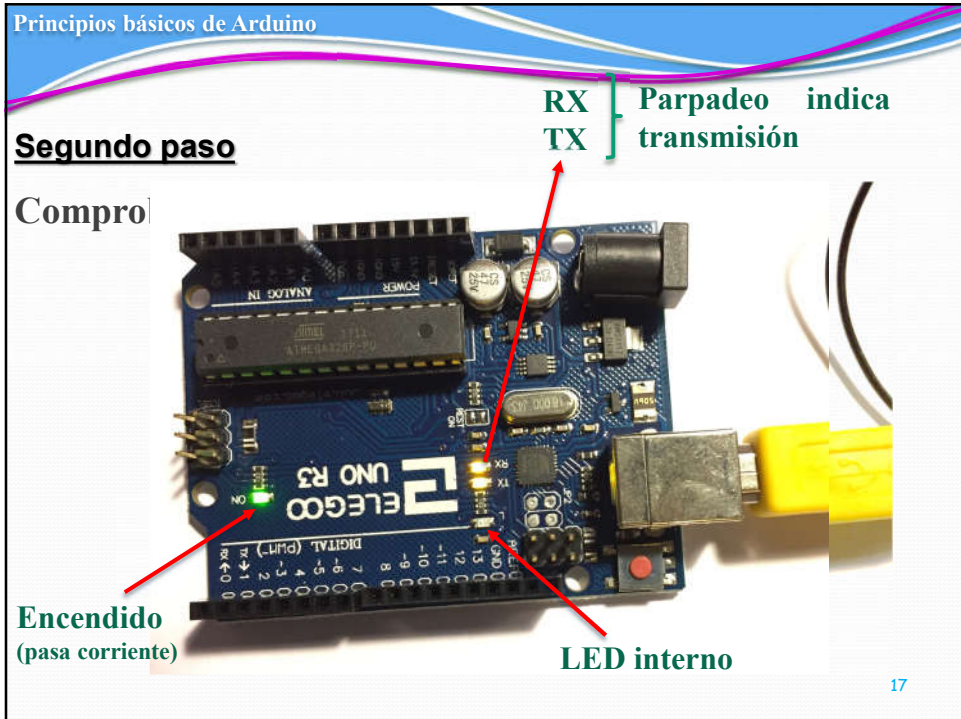




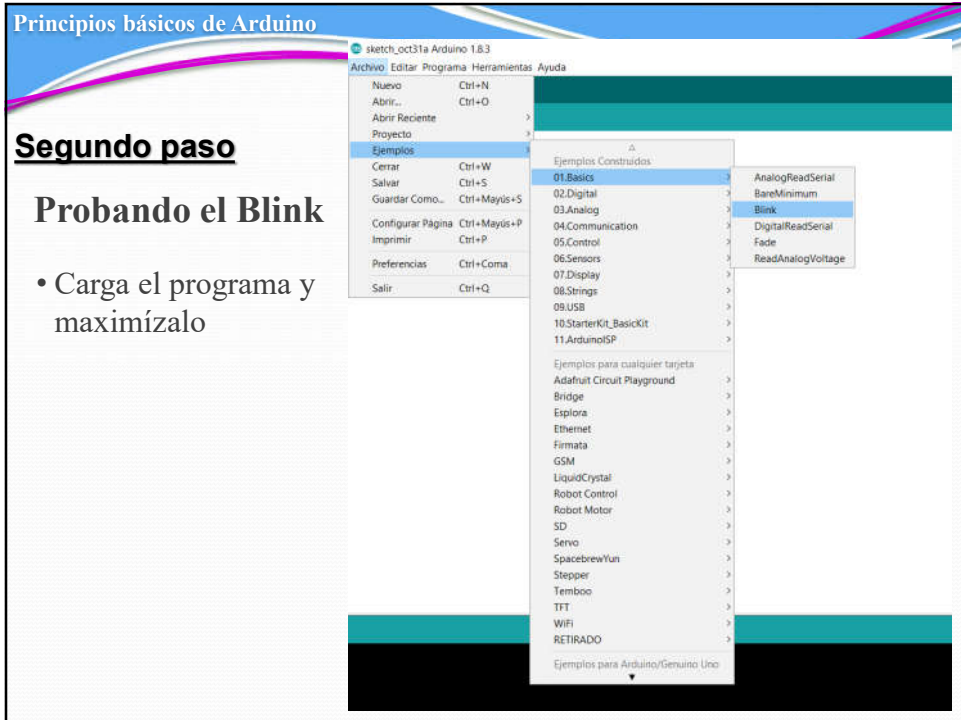
Arduino IDE 2.0.1

16

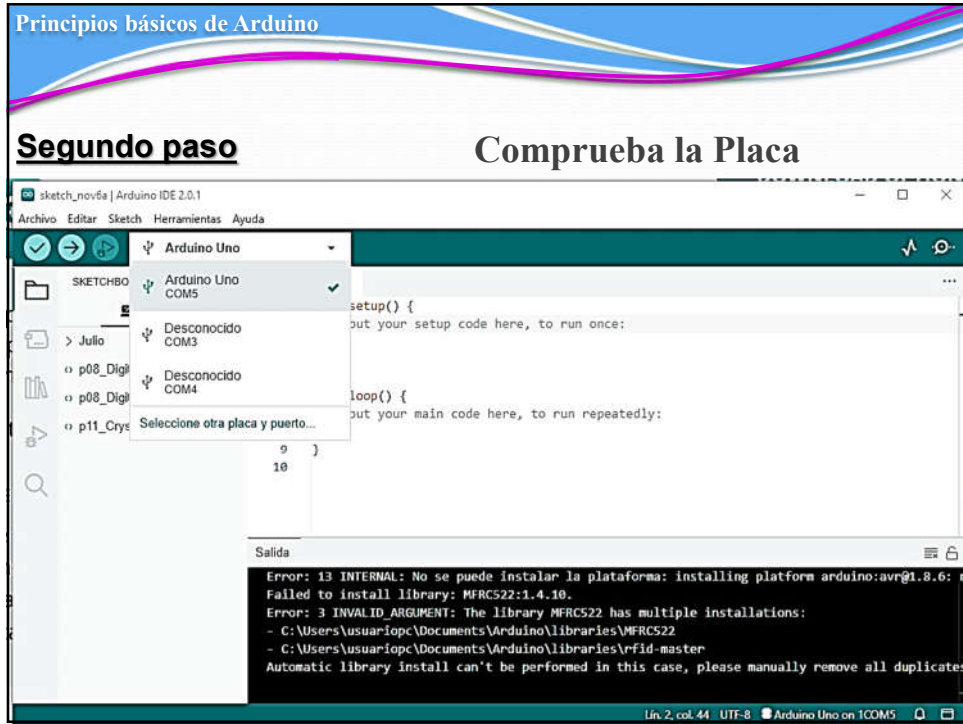




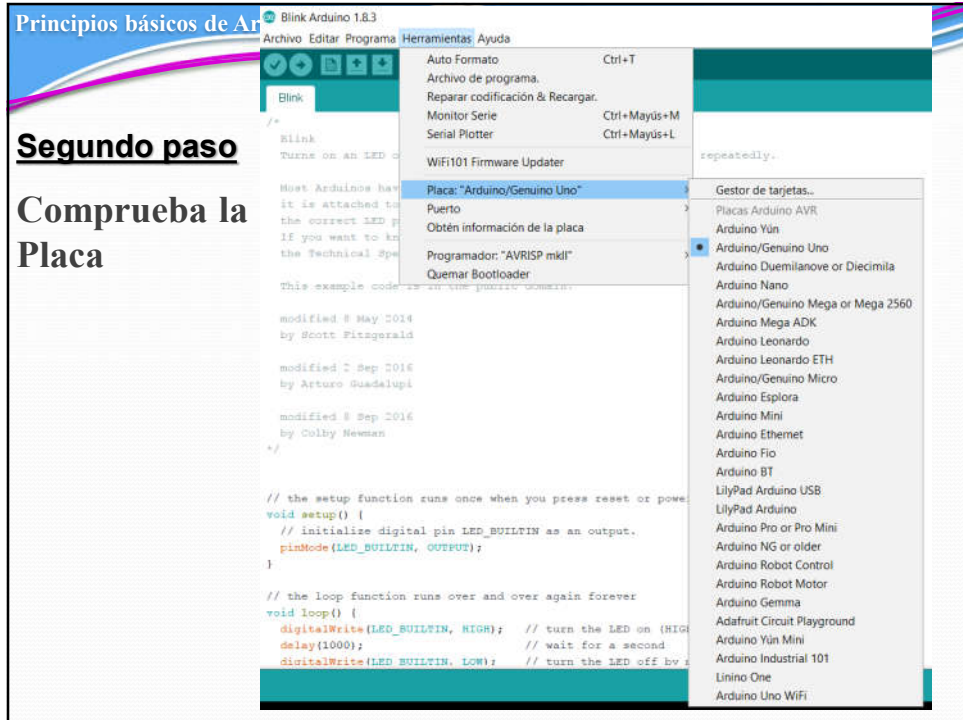
17



18



19



20

Principios básicos de Arduino

## Segundo paso

### Comprueba el puerto

The screenshot shows the Arduino IDE interface. The 'Tools' menu is open, and 'COM4 (Arduino/Genuino Uno)' is selected under the 'Puertos Serie' section. The code editor shows the 'Blink' sketch with the following code:

```

// Blink
// Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
it is attached to digital pin 13, on MEGA1000 on pin 6. LED_BUILTIN is set to
the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino model, check
the Technical Specs of your board at https://www.arduino.cc/en/Main/Products

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald

modified 2 Sep 2016
by Arturo Guadalupi

modified 8 Sep 2018
by Colby Newman
*/

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

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

21

Principios básicos de Arduino

## Segundo paso

### Cárgalo a la placa de Arduino

**Void setup**  
Que cosas tenemos y donde están conectadas

**Void loop**  
Que queremos que haga repetidamente

The screenshot shows the Arduino IDE interface with the 'Blink' sketch. Annotations explain the structure of the code:

- Encabezado:** "Autor, Fecha y lo que te dé la gana"
- Definición de variables:** (None explicitly shown in the code, but implied by the comment 'LED\_BUILTIN is set to the correct LED pin...')
- void setup:** "Configuración"
- void loop:** "Repetir"

The code in the editor is the same as in the previous slide:

```

// Blink
// Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
it is attached to digital pin 13, on MEGA1000 on pin 6. LED_BUILTIN is set to
the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino model, check
the Technical Specs of your board at https://www.arduino.cc/en/Main/Products

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald

modified 2 Sep 2016
by Arturo Guadalupi

modified 8 Sep 2018
by Colby Newman
*/

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

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

22

Principios básicos de Arduino

### Placa Arduino

## Segundo paso

### Comprobantes



Compilando programa...

Compilando

Arduino/Genuino Uno en COM4

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

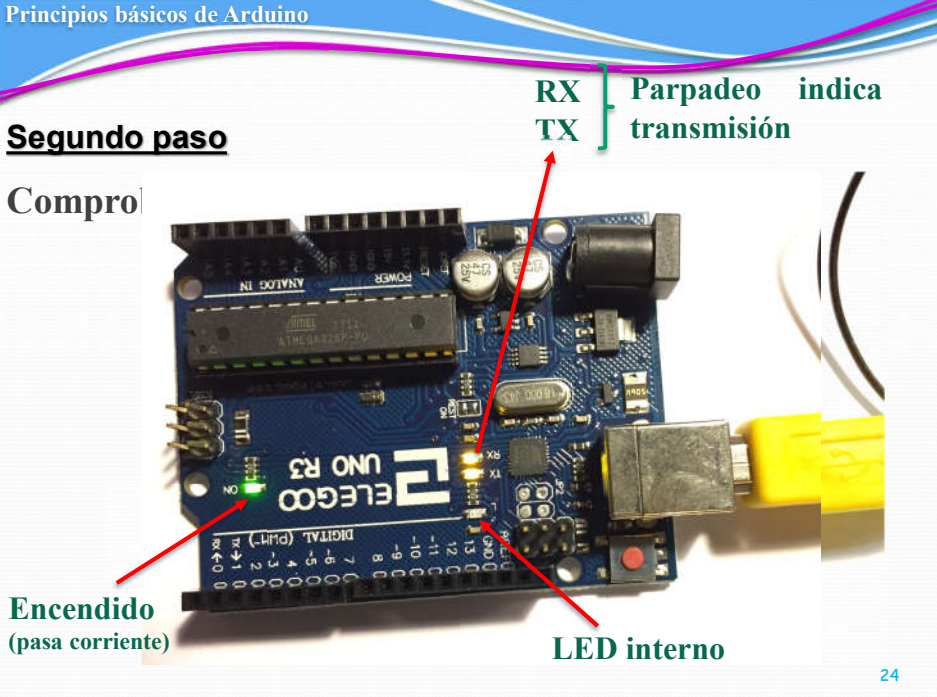
// Loop function that runs over and over again forever:
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED on (HIGH is the voltage level)
  delay(1000);                    // Wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED off by making the voltage LOW
}
```

23

Principios básicos de Arduino

## Segundo paso

### Compro



RX TX } Parpadeo indica transmisión

Encendido (pasa corriente)

LED interno

24



Principios básicos de Arduino

**TIN  
KER  
CAD**

**Segundo paso** *Programa que crea tu proyectos y los simula sin soporte físico*

Programas complementarios



The screenshot shows the TIN KER CAD web interface. On the left, there is a 3D model of an Arduino Uno board. On the right, a user menu is open for 'juliosci58DY9', listing options like 'Nuevo diseño', 'Mis diseños', 'Notificaciones', 'Perfil', 'Mis clases', and 'Cerrar sesión'. Below the board, there are 'ODESK' and 'de menores >' labels.

27

Principios básicos de Arduino

**TIN  
KER  
CAD**

**Segundo paso** *Programa que crea tu proyectos y los simula sin soporte físico*

Programas complementarios



The screenshot shows the TIN KER CAD web interface with a block-based programming environment. On the left, there is a 3D model of an Arduino Uno board. On the right, a block-based code editor is visible, showing a sequence of blocks for controlling an LED, including 'Encendido de LED', 'Apagado de LED', and 'Encendido de LED'. The interface includes a toolbar at the top and a 'Monitor de serie' button at the bottom.

28

Principios básicos de Arduino

**TIN**  
**KER**  
**CAD**

**Segundo paso** *Programa que crea tu proyectos y los simula sin soporte físico*

### Programas complementarios

```
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

29

Principios básicos de Arduino

**TIN**  
**KER**  
**CAD**

**Segundo paso**

### Programas complementarios

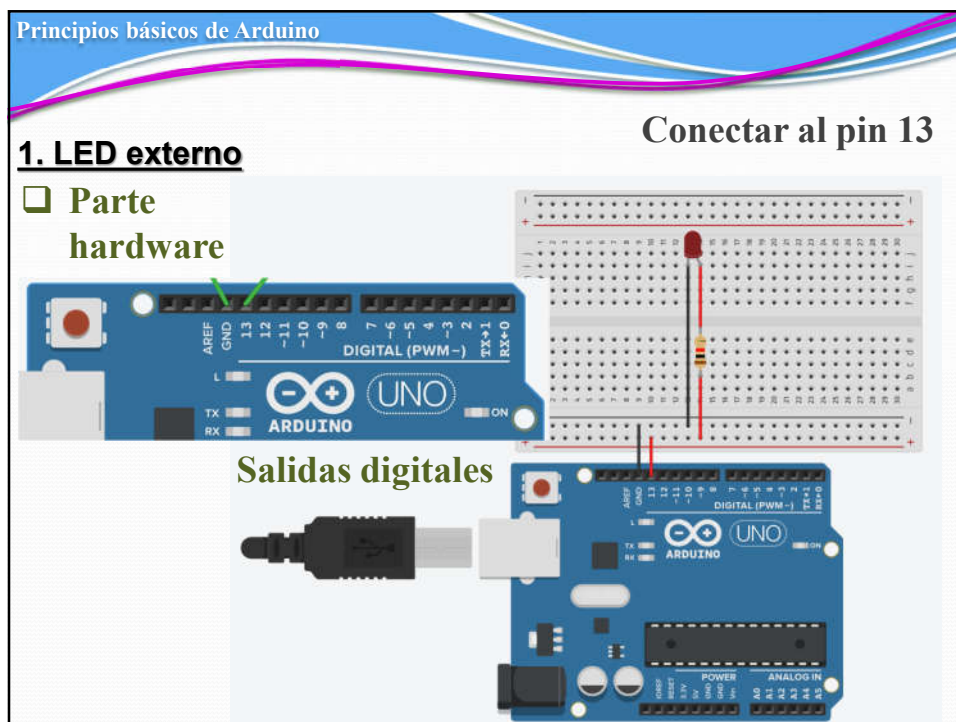
**Algoritmos**

Un algoritmo es una serie de instrucciones sencillas, paso a paso, que en conjunto ejecutan una tarea concreta. En este proyecto hemos convertido la coreografía de la bailarina (un algoritmo) en un programa. Todos los programas informáticos tienen, en su corazón, un algoritmo. Programar es traducir los pasos del algoritmo a un lenguaje de programación que el ordenador comprenda.

ALGORITMO (pasos de baile)	PROGRAMA (pasos de baile convertidos en lenguaje de programación)
¡DE PIE!	cambiar disfraz a (ballerina-a)
¡ESPERA!	esperar 0.5 segundos
¡MUEVE LAS PIERNAS!	cambiar disfraz a (ballerina-d)
¡ESPERA!	esperar 0.5 segundos
¡DE PIE!	cambiar disfraz a (ballerina-a)
¡ESPERA!	esperar 0.5 segundos
¡AGÁCHATE!	cambiar disfraz a (ballerina-b)

30

30



31

Principios básicos de Arduino

### 1. LED externo

```
// C++ code
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

32

32



Principios básicos de Arduino

**1. LED externo** ¡Prueba tu mismo!

Realiza el mismo montaje anterior pero utilizando el pin 8 para conectar el diodo LED.

- 1 Con el montaje anterior haz que el LED esté encendido 1 segundo y apagado 2.
- 2 Diseña un montaje con dos LED conectados a los pines digitales 9 y 4. El primer LED deberá encenderse y apagarse cada 4 segundos. El segundo LED cada 3 segundos.

33

33

Principios básicos de Arduino

Realiza el mismo montaje anterior pero utilizando el pin 8 para conectar el diodo LED.

**1. LED**

- 1 Con el montaje anterior haz que el LED esté encendido 1 segundo y apagado 2.

```

http://www.arduino.cc/en/Tutorial
*/
int LED = 8;

// the setup function runs once when the board is first powered up
void setup() {
  // initialize digital pin LED_BUILTIN as an output:
  pinMode(LED, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000);              // wait for a second
  digitalWrite(LED, LOW);  // turn the LED off by making the pin LOW
  delay(2000);             // wait for two seconds
}
    
```

34

34

Principios básicos de Arduino

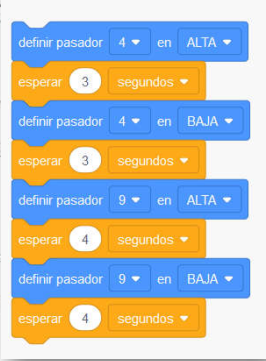
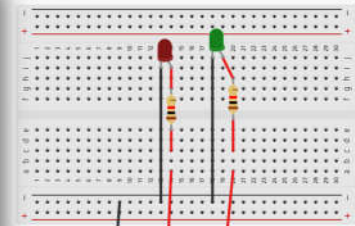
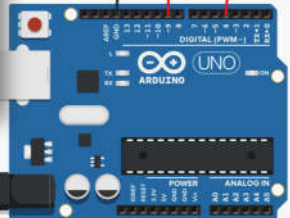
**1. LEI** Diseña un montaje con dos LED conectados a los pines digitales 9 y 4. El primer LED deberá encenderse y apagarse cada 4 segundos. El segundo LED cada 3 segundos.

**2**

```

http://www.arduino.cc/en/Tutor
/
nt LED_VERDE = 9;
nt LED_ROJO = 4;
/ the setup function runs once
oid setup() {
  // initialize digital pin LED
  pinMode(LED_VERDE, OUTPUT);
  pinMode(LED_ROJO, OUTPUT);

/ the loop function runs over a
oid loop() {
  digitalWrite(LED_VERDE, HIGH);
  delay(4000);
  digitalWrite(LED_VERDE, LOW);
  delay(4000);
  digitalWrite(LED_ROJO, HIGH);
  delay(3000);
  digitalWrite(LED_ROJO, LOW);
  delay(3000);
  // wait for a second
  // wait for a second
  
```

35

Principios básicos de Arduino

**2. Nos iniciamos en la programación**

- *¡Construye El coche fantástico (1982)*

**Diseño**






36

Principios básicos de Arduino

## 2. Nos iniciamos en la programación

- *Construye El coche fantástico (1982)*

**Diseño**



37

Principios básicos de Arduino

## 2. Nos iniciamos en la programación

- *Construye El coche fantástico*

**Código**

```
void setup() {
  // put your setup code
  here, to run once:
  pinMode (2, OUTPUT);
  pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (6, OUTPUT);
}
```

```
void loop() {
  digitalWrite (2, HIGH);
  delay(1000);
  digitalWrite (2, LOW);
  digitalWrite (3, HIGH);
  delay(300);
  digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  delay(300);
  digitalWrite (4, LOW);
  digitalWrite (5, HIGH);
  delay(300);
  digitalWrite (5, LOW);
  digitalWrite (6, HIGH);
  delay(300);
  digitalWrite (6, LOW);
  digitalWrite (5, HIGH);
  delay(300);
  digitalWrite (5, LOW);
  digitalWrite (4, HIGH);
  delay(300);
  digitalWrite (4, LOW);
  digitalWrite (3, HIGH);
  delay(300);
  digitalWrite (3, LOW);
}
```

38

38

Principios básicos de Arduino

## 2. Nos iniciamos en la programación

- Construye El coche fantástico (1982)

**Código**

**T I N**

**K E R**

**C A D**

definir pasador 2 en ALTA

esperar 1 segundos

definir pasador 2 en BAJA

definir pasador 3 en ALTA

esperar 0.3 segundos

definir pasador 3 en BAJA

definir pasador 4 en ALTA

esperar 0.3 segundos

definir pasador 4 en BAJA

definir pasador 5 en ALTA

esperar 0.3 segundos

definir pasador 5 en BAJA

definir pasador 6 en ALTA

esperar 0.3 segundos

definir pasador 6 en BAJA

definir pasador 5 en ALTA

esperar 0.3 segundos

definir pasador 5 en BAJA

definir pasador 4 en ALTA

esperar 0.3 segundos

definir pasador 4 en BAJA

definir pasador 3 en ALTA

esperar 0.3 segundos

definir pasador 3 en BAJA

39

39


Principios básicos de Arduino

## 2. Nos iniciamos en la programación


- ¡Construye tu nave espacial!

5.1. Monta el circuito

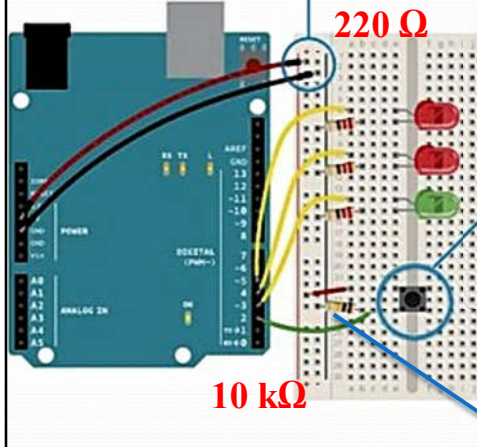
El pulsador y los LEDs se encuentran controlados por *Arduino*: la información e intensidad sale de la placa digital



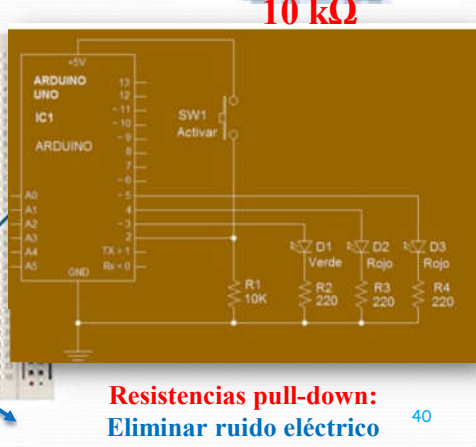
220 Ω



10 kΩ



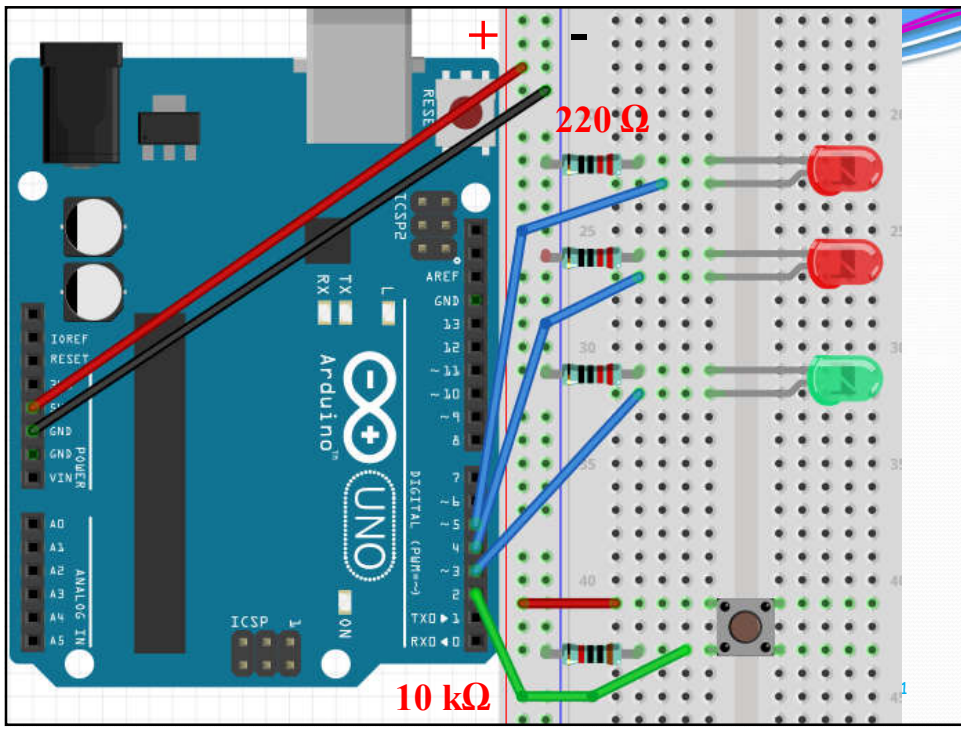
10 kΩ



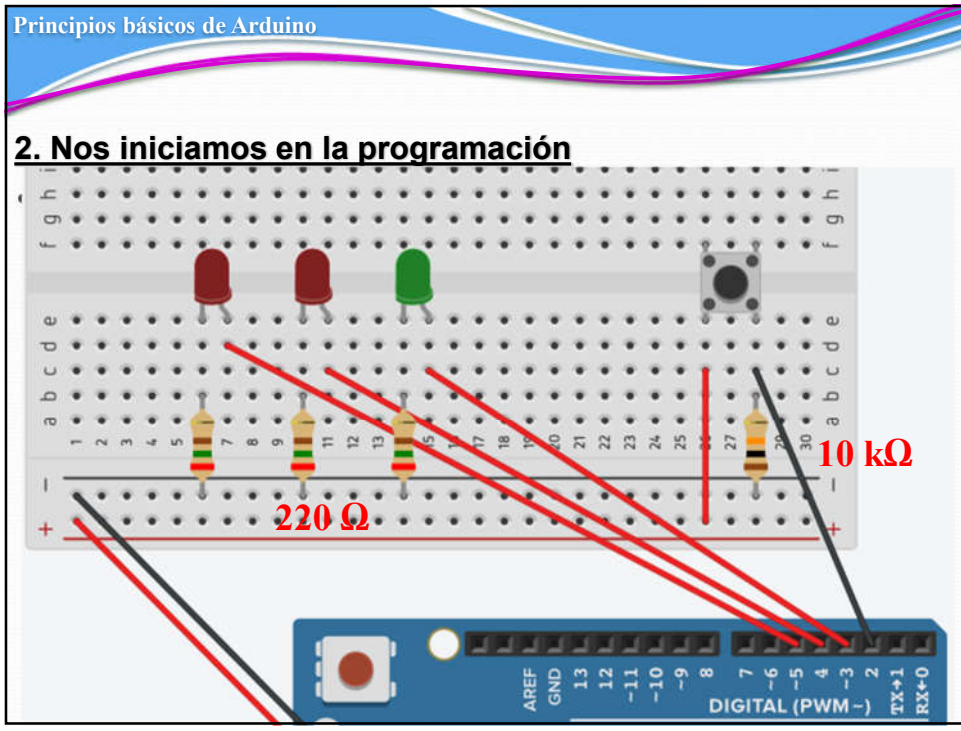
Resistencias pull-down:  
Eliminar ruido eléctrico

40

40

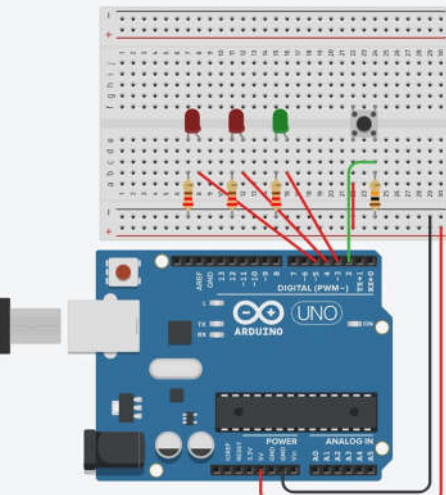


41



42

**TIN** Principios de Arduino  
**KER**  
**CAD** vamos en la programación



comentario de bloque de título Programa nave espacial con tres led

comentario Condicional para trabajar

si leer pasador digital 2 = BAJA entonces

- definir pasador 3 en ALTA
- definir pasador 4 en BAJA
- definir pasador 5 en BAJA

si no

- definir pasador 3 en BAJA
- definir pasador 4 en BAJA
- definir pasador 5 en ALTA
- esperar 25 segundos
- definir pasador 4 en ALTA
- definir pasador 5 en BAJA
- esperar 25 segundos

43

Principios básicos de Arduino

## 2. Nos iniciamos en la programación

- ¡Construye tu nave espacial!

### 5.2. Programemos

#### 2.7 Retos

1. Haz parpadear el verde sin pulsar el interruptor

¡¡No funciona!!

comentario de bloque de título Programa nave espacial con tres led

comentario Condicional para trabajar

si leer pasador digital 2 = BAJA entonces

- definir pasador 3 en BAJA
- definir pasador 4 en BAJA
- definir pasador 5 en BAJA
- esperar 1 segundos
- definir pasador 3 en ALTA

si no

- definir pasador 3 en BAJA
- definir pasador 4 en BAJA
- definir pasador 5 en ALTA
- esperar 25 segundos
- definir pasador 4 en ALTA
- definir pasador 5 en BAJA
- esperar 25 segundos

44



comentario Condicional para trabajar

si leer pasador digital 2 = BAJA entonces

- definir pasador 3 en ALTA
- definir pasador 4 en BAJA
- definir pasador 5 en BAJA

si no

- esperar 5 segundos
- definir pasador 3 en BAJA
- definir pasador 4 en ALTA

esperar 1 segundos

- definir pasador 4 en BAJA
- definir pasador 5 en ALTA

esperar 10 segundos

47

47

Principios básicos de Arduino

**Reto:**  
Que parpadee el verde al final dos veces antes de pasar a rojo

5.3. Monta el circuito (ya mo

si leer pasador digital 2 = BAJA entonces

- definir pasador 5 en ALTA
- definir pasador 4 en BAJA
- definir pasador 3 en BAJA

si no

- esperar 5 segundos
- definir pasador 5 en BAJA
- definir pasador 4 en ALTA
- definir pasador 3 en BAJA

esperar 2 segundos

- definir pasador 4 en BAJA
- definir pasador 3 en ALTA

esperar 3 segundos

- definir pasador 3 en BAJA

esperar 1 segundos

- definir pasador 3 en ALTA

esperar 1 segundos

- definir pasador 3 en BAJA

esperar 1 segundos

- definir pasador 3 en ALTA

esperar 1 segundos

48



Principios básicos de Arduino

**Reto:**  
Que parpadee el verde al final dos veces antes de pasar a rojo

2. Crea tu sensorio:

5.3. Monta el circuito (ya mo

49

Principios básicos de Arduino

**3. Los sensores**

• La máquina del amor *EL ADC*

- Clavijas de A0 hasta A5
  - *Rango de valores:* 0-1023
  - *Voltaje:* de 0 a 5
- Sensor de temperatura (analógicos)

El conversor Analógico Digital (ADC) para sensores

50

Principios básicos de Arduino

### 3. Los sensores

- La máquina del amor

#### 6.1. Monta el circuito

5 V (A0) GND

Figura 3

220  $\Omega$

51

51

Principios básicos de Arduino

### 3. Los sensores

- La máquina del amor

#### 6.1. Monta el circuito

5 V (A0) GND

220  $\Omega$

52

52

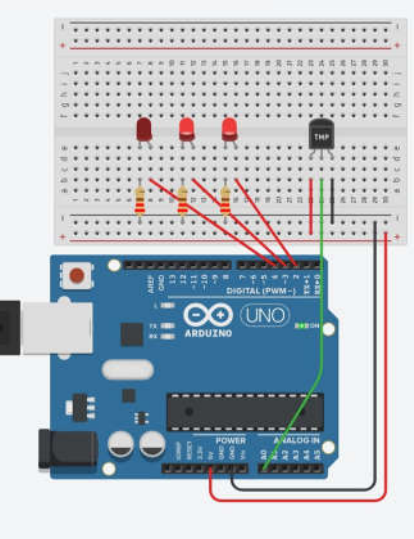


Principios básicos de Arduino

**TIN  
KER  
CAD**

Meter el programa en TINKERCAD

### 3. Los sensores



```

19 // named constant for the pin the sensor is connected to
20 const int sensorPin = A0;
21 // room temperature in Celsius
22 const float baselineTemp = 20.0;
23
24
25 void setup() {
26   // open a serial connection to display values
27   Serial.begin(9600);
28   // set the LED pins as outputs
29   // the for() loop saves some extra coding
30   for (int pinNumber = 2; pinNumber < 5; pinNumber++) {
31     pinMode(pinNumber, OUTPUT);
32     digitalWrite(pinNumber, LOW);
33   }
34 }
35
36 void loop() {
37   // read the value on Analog pin 0 and store it in a variable
38   int sensorVal = analogRead(sensorPin);
39
40   // send the 10-bit sensor value out the serial port
41   Serial.print("sensor Value: ");
42   Serial.println(sensorVal);
43
44   // convert the ADC reading to voltage
45   float voltage = (sensorVal / 1024.0) * 5.0;
46
47   // Send the voltage level out the Serial port
48   Serial.print("V: ");
49   Serial.println(voltage);
50
51   // convert the voltage to temperature in degrees C
52   // the sensor changes 10 mV per degree
53   // the datasheet says there's a 500 mV offset
54   // (Voltage - 500 mV) / (10 mV)
55   Serial.print("Temperature: ");
56 }

```

Monitor en serie

```

24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71
24.71sensor Value: 153, Voltaje: 0.75, grados C: 24.71

```

Env. Borrar

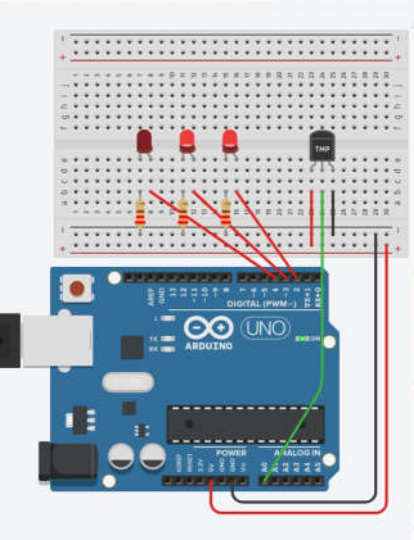
55

Principios básicos de Arduino

**TIN  
KER  
CAD**

Crear el programa con TINKERCAD

### 3. Los sensores



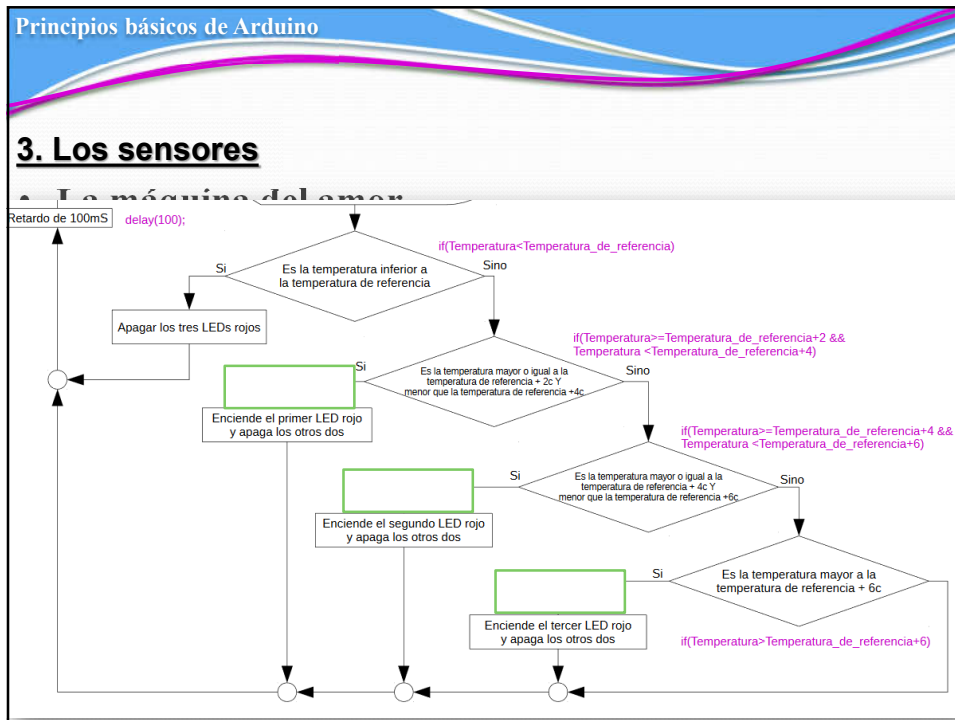
```

1 Leer el sensor de temperatura en el pasador A0 en las unidades °C 20 unidades
define pasador 2 en BAA
define pasador 3 en BAA
define pasador 4 en BAA
2 Leer el sensor de temperatura en el pasador A0 en las unidades °C 20 2 unidades
define pasador 2 en ALA
define pasador 3 en BAA
define pasador 4 en BAA
3 Leer el sensor de temperatura en el pasador A0 en las unidades °C 20 2 unidades
define pasador 2 en ALA
define pasador 3 en BAA
define pasador 4 en BAA
4 Leer el sensor de temperatura en el pasador A0 en las unidades °C 20 3 unidades
define pasador 2 en ALA
define pasador 3 en ALA
define pasador 4 en ALA
5 Leer el sensor de temperatura en el pasador A0 en las unidades °C 20 3 unidades
define pasador 2 en ALA
define pasador 3 en ALA
define pasador 4 en ALA
6 Leer el sensor de temperatura en el pasador A0 en las unidades °C 20 3 unidades
define pasador 2 en ALA
define pasador 3 en ALA
define pasador 4 en ALA

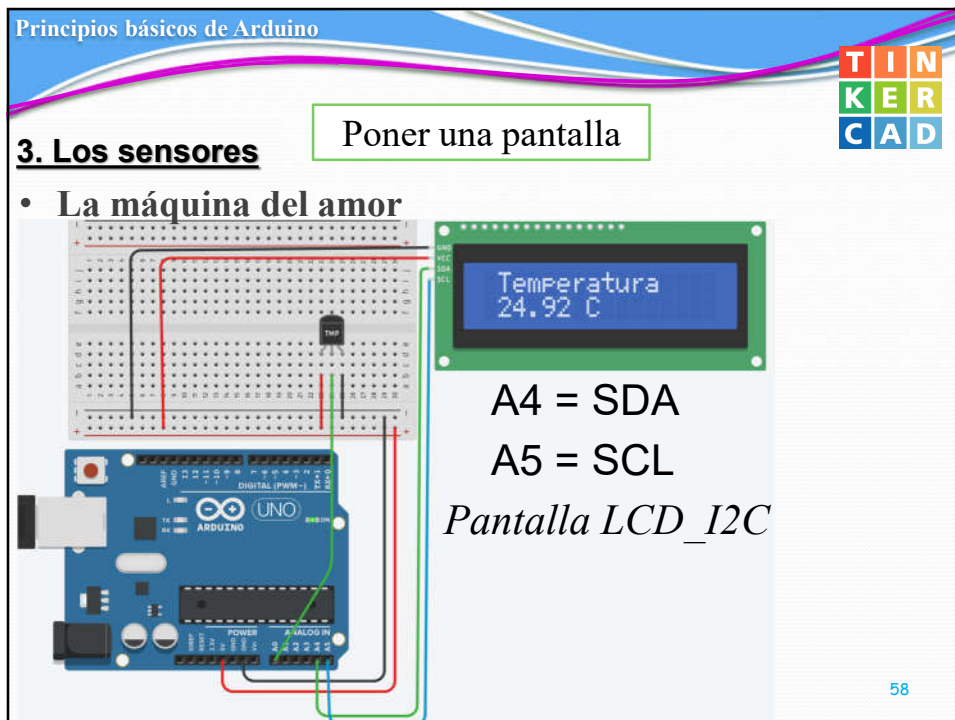
```

Imprimir en monitor en serie Leer el sensor de temperatura en el pasador A0 en las unidades °C Nueva línea con

56



57



58

58

Principios básicos de Arduino

T I N  
K E R  
C A D

al iniciar

configurar tipo de LCD 1 en I2C (PCF8574) con dirección 32

imprimir en monitor en serie leer el sensor de temperatura en el pasador A0 en las unidades °C , nueva línea con

en LCD 1 activar la luz de fondo

establecer posición en LCD 1 en columna 1 , fila 0

imprimir en LCD 1 Temperatura

establecer posición en LCD 1 en columna 1 , fila 1

imprimir en LCD 1 leer el sensor de temperatura en el pasador A0 en las unidades °C

establecer posición en LCD 1 en columna 7 , fila 1

imprimir en LCD 1 C

59

Principios básicos de Arduino

RGB\_LED\_Julio Arduino 1.8.10

Archivo Editar Programa Herramientas Ayuda

Verificar/Compilar Ctrl+R

Subir Ctrl+U

Subir Usando Programador Ctrl+Mayús+U

Exportar Binarios compilados Ctrl+Alt+S

Mostrar Carpeta de Programa Ctrl+K

Incluir Librería

Añadir fichero...

Administrar Bibliotecas... Ctrl+Mayús+I

Añadir biblioteca ZIP...

Arduíno bibliotecas

Bridge

EEPROM

Esplora

Ethernet

Firmata

GSM

HID

Keyboard

LiquidCrystal

Mouse

Robot Control

Robot IR Remote

Robot Motor

SD

SPI

Servo

SoftwareSerial

Spacebrew\un

Stepper

TFT

Temboo

WiFi

Wire

Contribución bibliotecas

Arduíno-LiquidCrystal-I2C-library-master

Average-master

DS1302

### 3. Los sensores

- La máquina del am

Carga la librería  
“*Arduino-LiquidCrystal-I2C-library-master*” de la plataforma

```
//Www.e
//2016.12.8

// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6

void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH); // ES
  digitalWrite(GREEN, HIGH); //
  digitalWrite(BLUE, HIGH); // E
}
```

60

Principios básicos de Arduino

### 3. Los sensores

- La máquina del amor

Carga el programa “*Love meter con pantalla*” de la plataforma

#### Declarar la librería de la pantalla

```
#include <Average.h>

#include <LiquidCrystal_I2C.h> //
Incluimos la librería del modulo LCD
con el adaptador disminuidor de
pines

// Creación de objetos LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

//Hasta aquí hemos declarado la
librería para la LCD y los pines por
donde le va a entrar la información.
```

61

61

Principios básicos de Arduino

### 3. Los sensores

- La máquina del amor

Carga el programa “*Love meter con pantalla*” de la plataforma

```
float centi()
{// Funcion para leer el dato analógico y
convertirlo a digital:

int dato;
int ave;
float c;
dato=analogRead(A0);
dato=ave;
c = (500.0 * dato)/1023-50;

// relación del sensor con los grados
return (c);
}

float kelvin(float cent)
{
float k;
k=cent+273.15;
return(k);

// Convertir centígrados a kelvin
}
```

62

62

Principios básicos de Arduino

### 3. Los sensores

- **La máquina del amor**

Carga el programa “*Love meter con pantalla*” de la plataforma. Sin embargo, la pantalla no funciona correctamente (fórmula no bien programada)

```

void setup() {
  // Definimos la LCD con dimension 2x16 y
  // definimos los caracteres que deben salir en
  // las filas:
  // Iniciar pantalla
  lcd.begin();
  lcd.backlight();//Retroiluminar pantalla
  lcd.print("C=      K=");
  lcd.setCursor(0,1);
  lcd.print("Temperatura");

  Average<float> ave(10);
}

// Escribir los datos:
float Centigrados = centi();
float Kelvin = kelvin (Centigrados);

lcd.setCursor(2,0);
lcd.print(Centigrados);
lcd.setCursor(10,0);
lcd.print(Kelvin);
delay(600);
    
```

63

63

220  $\Omega$

### 3. Los sensores

- La bola de

6.3. Monta el circuito

10 k $\Omega$

Tilt ball switch

220  $\Omega$

fritzing

64



Principios básicos de Arduino

### Sexto paso: Los sensores

- La bola de cristal

6.3. Monta el circuito

Carga el programa “bola\_magica\_LCD\_I2C” de la plataforma o de la carpeta compartida

65

Principios básicos de Arduino

### 3. Los sensores

- La bola de cristal

6.3. Monta el circuito

Carga la librería “Arduino-LiquidCrystal-I2C-library-master” de la plataforma si no aparece en el ordenador

```

//RGB_LED_6
//2016.12.8

// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6

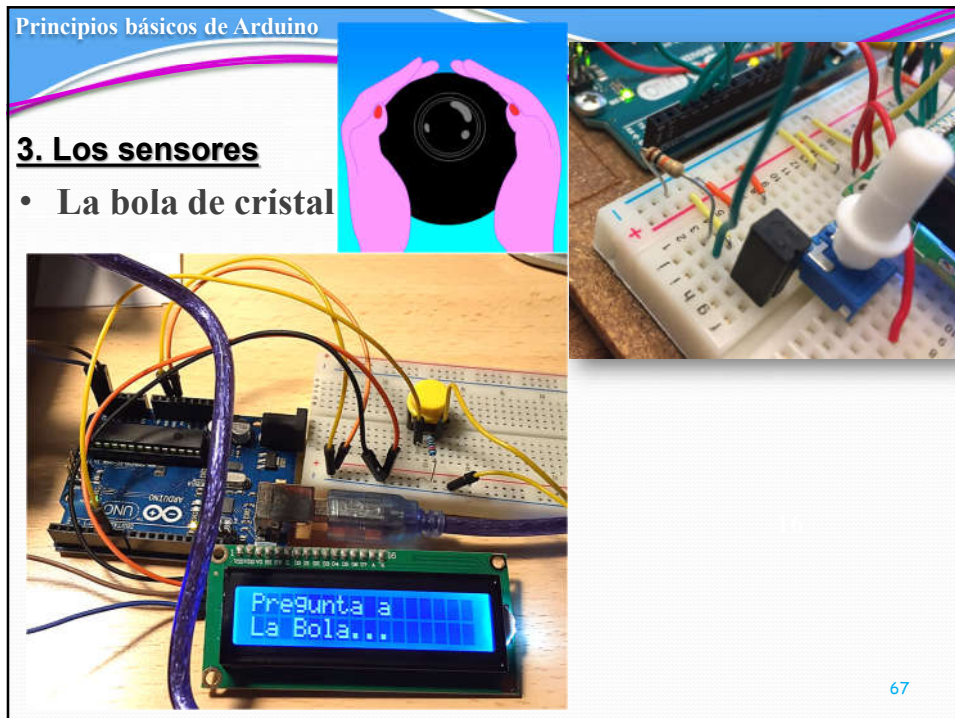
void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH); // Encendido
  digitalWrite(GREEN, HIGH); // Encendido
  digitalWrite(BLUE, HIGH); // Encendido
}
    
```

66

Principios básicos de Arduino

### 3. Los sensores

- La bola de cristal



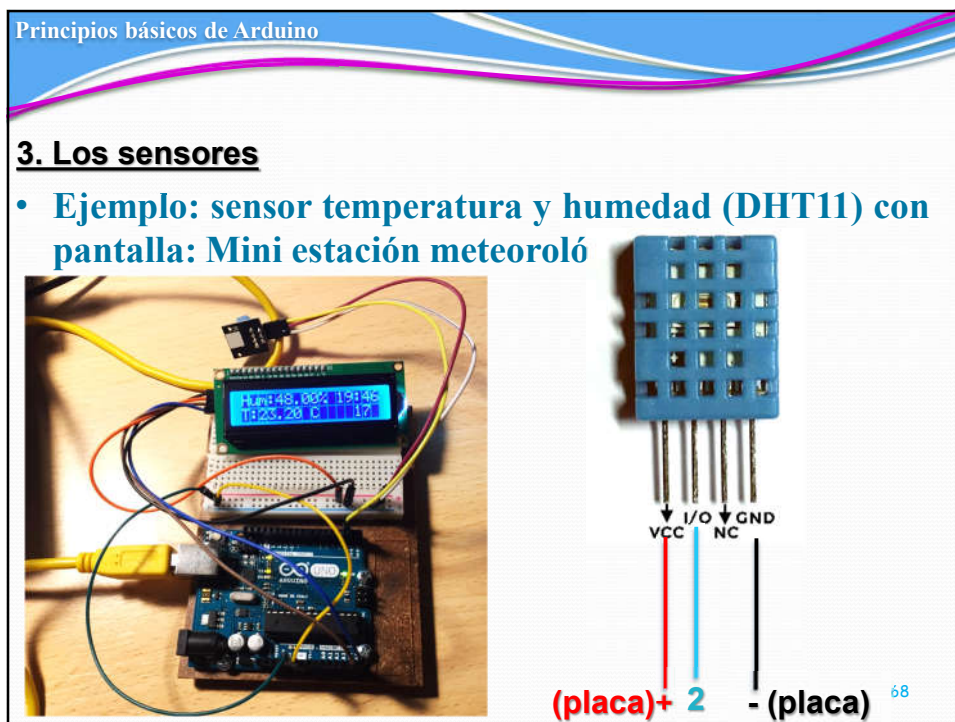
67

67

Principios básicos de Arduino

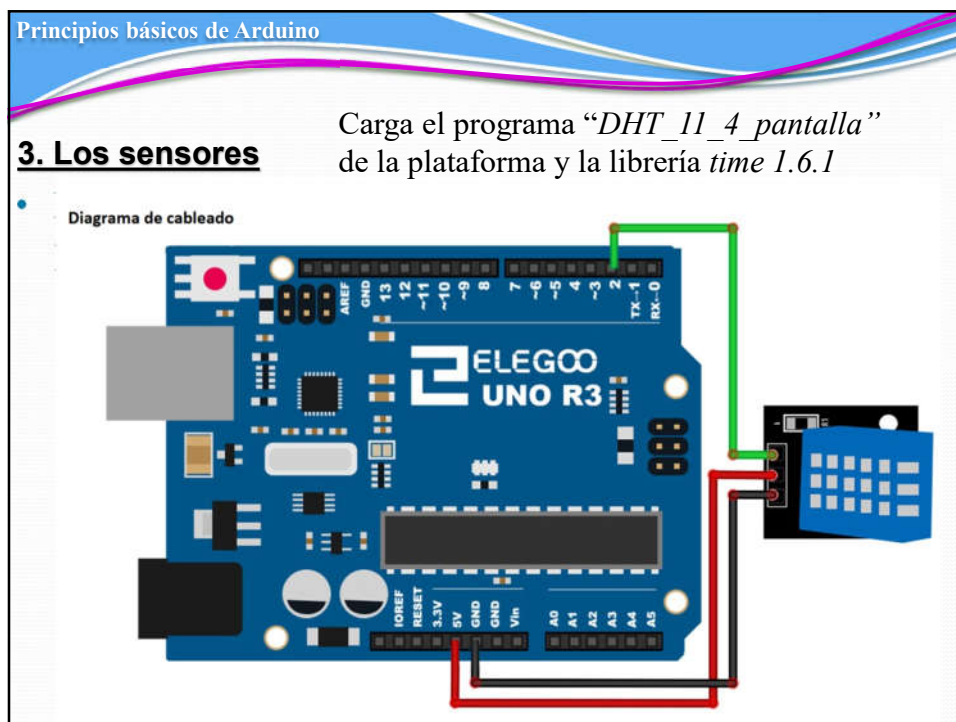
### 3. Los sensores

- Ejemplo: sensor temperatura y humedad (DHT11) con pantalla: Mini estación meteoroló



68

68



69

Principios básicos de Arduino

### 3. Los sensores

- La bola de cristal

6.3. Monta el circuito

Carga la librería “*DHT-sensor-library-master*” de la plataforma

Carga la librería “*Time 1.6.1.zip*” de la plataforma

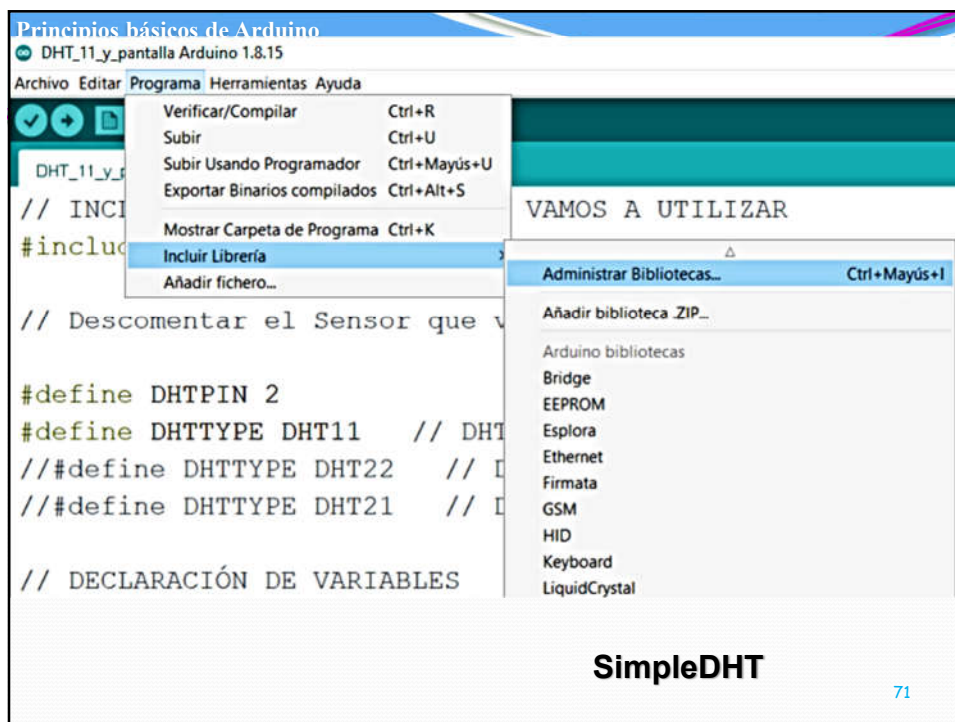
```

//Www.e
//2016.12.8

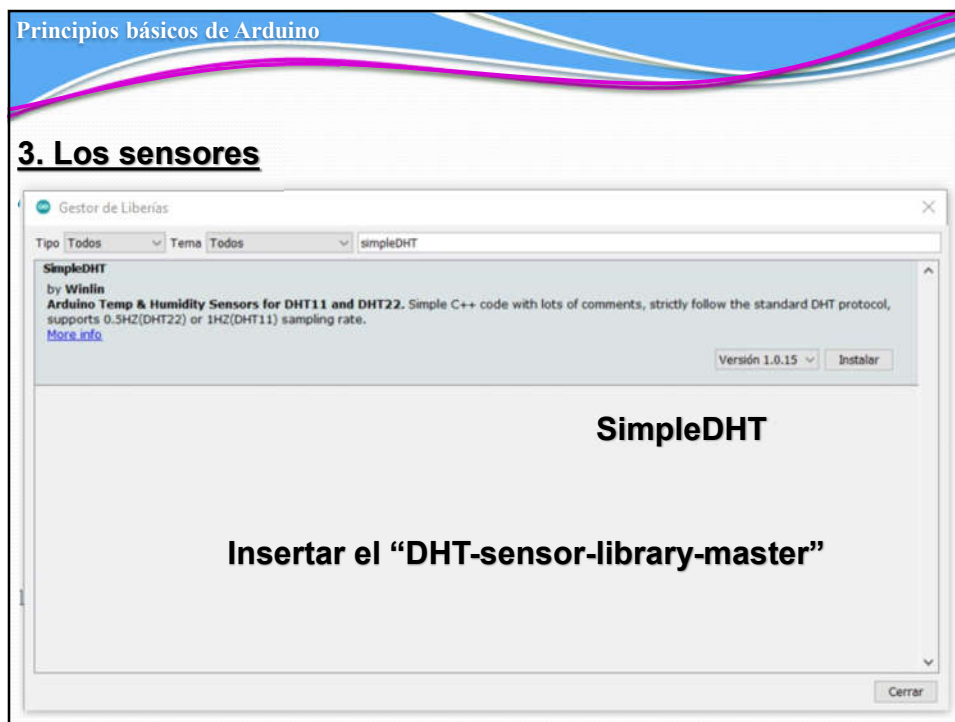
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6

void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH); // ES
  digitalWrite(GREEN, HIGH); // E
  digitalWrite(BLUE, HIGH); // E
}
    
```

70



71



72

Principios básicos de Arduino

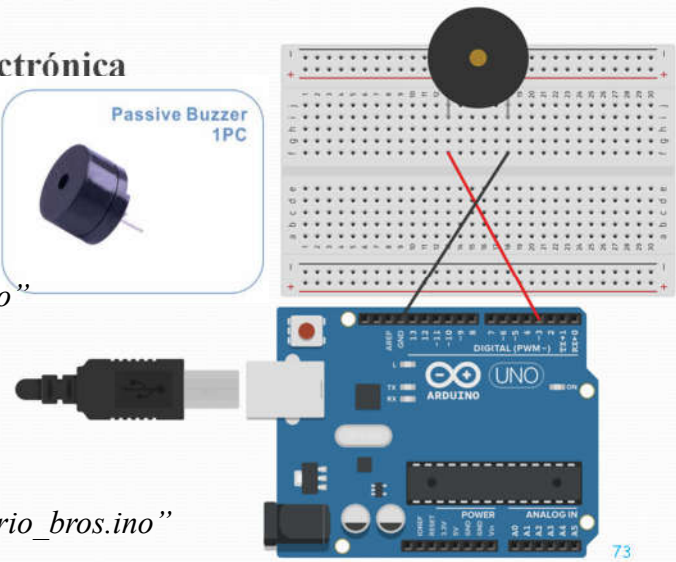
### 3. Los sensores

- **La música electrónica**

6.3. Monta el circuito

Carga el programa "musica\_popurri.ino" de la plataforma

Carga el programa "Ejercicio\_3\_1\_mario\_bros.ino" de la plataforma



Passive Buzzer 1PC

73

73

Principios básicos de Arduino

## El Arduino en la enseñanza

*Electrónica y didáctica*



74

74