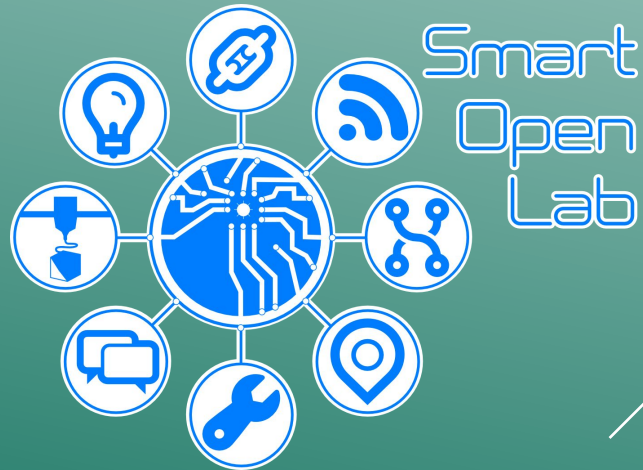




CURSO ARDUINO BÁSICO



JORGE BOTE ALBALÁ

 @rjoriks

 @joriks

MARÍA SALGUERO GALLEGO

 @mariesalguerog
 @mariesalguerog

¿QUÉ ES ARDUINO?

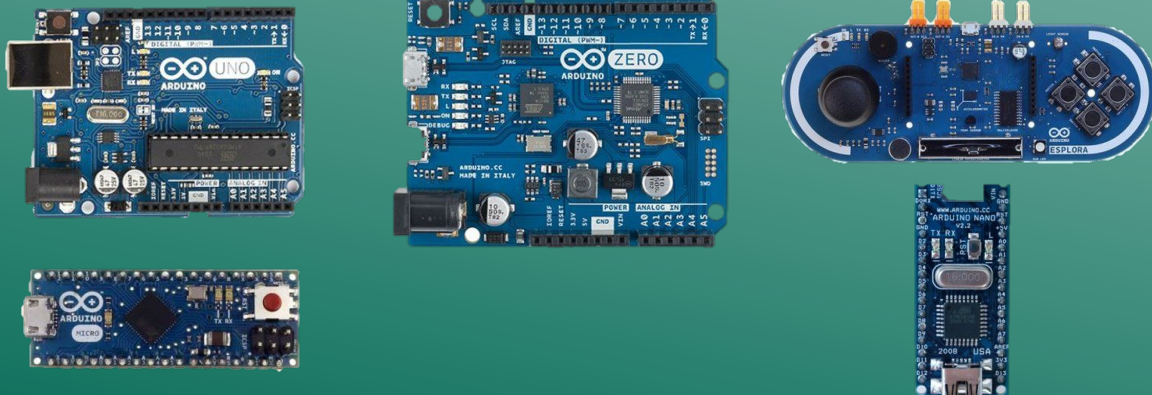
TOTAL PINES

GUARDA PROGRAMA

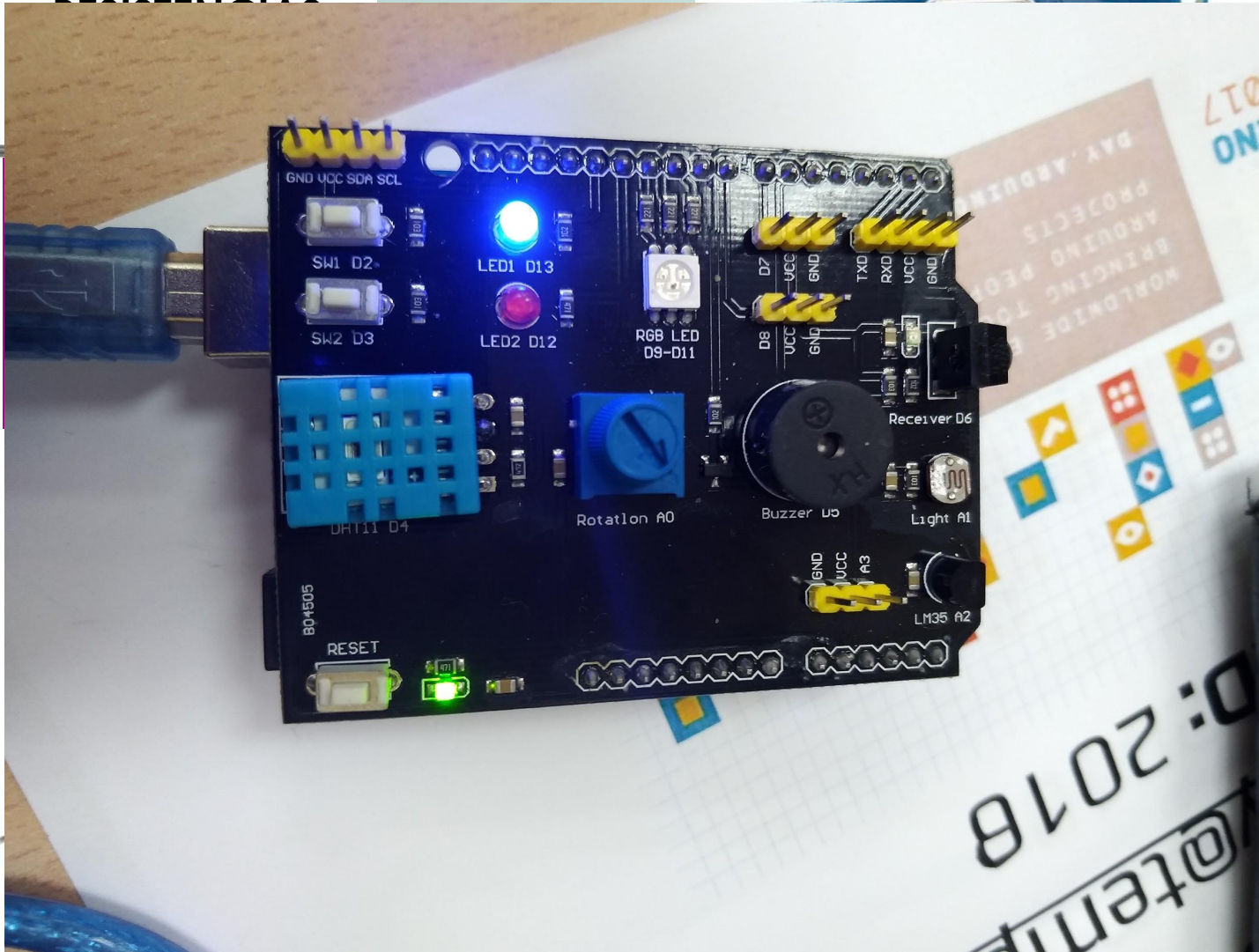
MEM. TOTAL

GUARDA DATOS

Arduino Uno	
Digital I/O Pins	14 (6 PWM)
Analog Input Pins	6
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Voltaje	5V



COMPONENTES QUE VAMOS A USAR (en este curso)

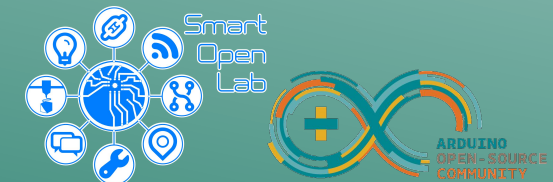


CABLE USB



ARDUINO UNO

RES

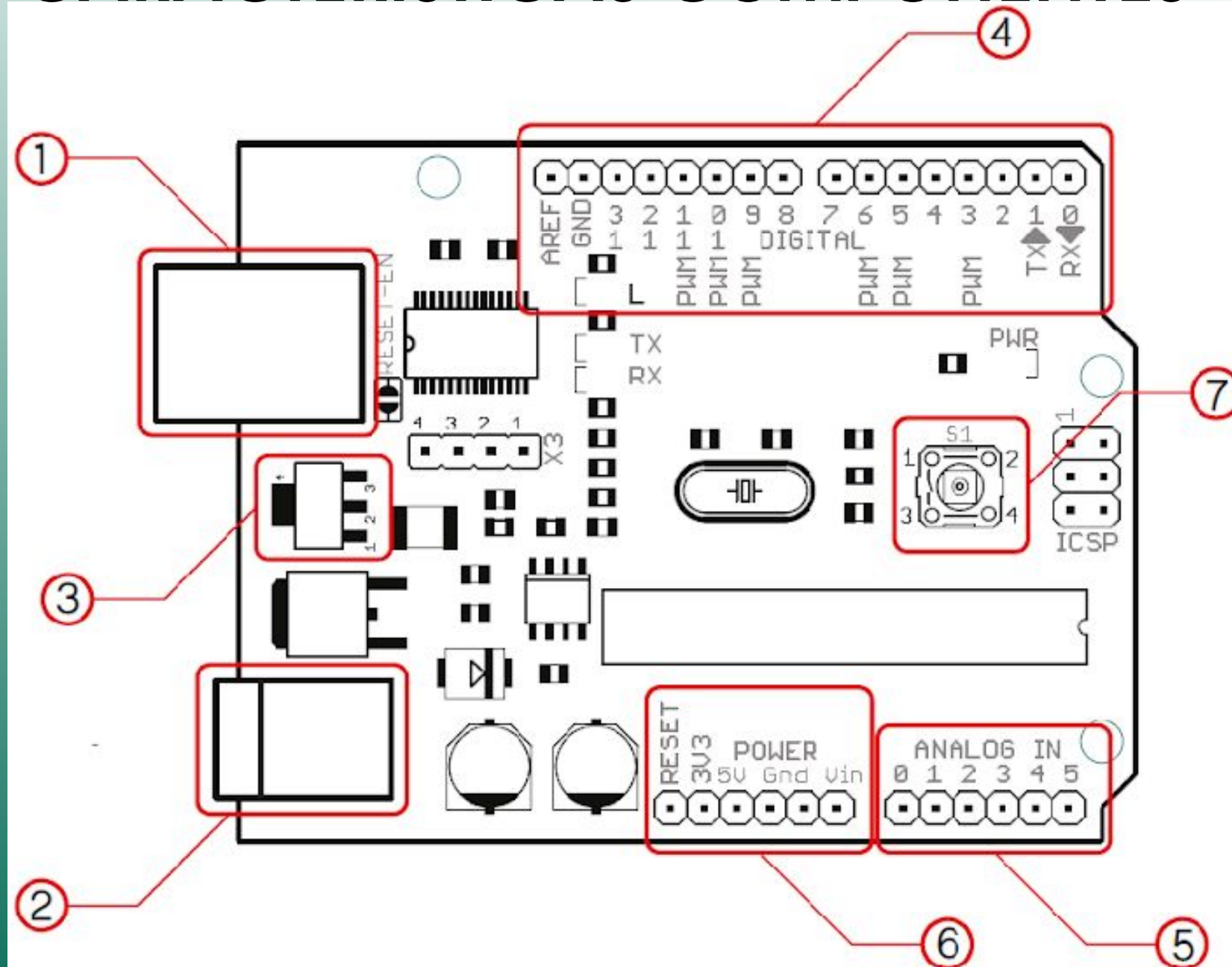


CARACTERÍSTICAS COMPONENTES

ARDUINO

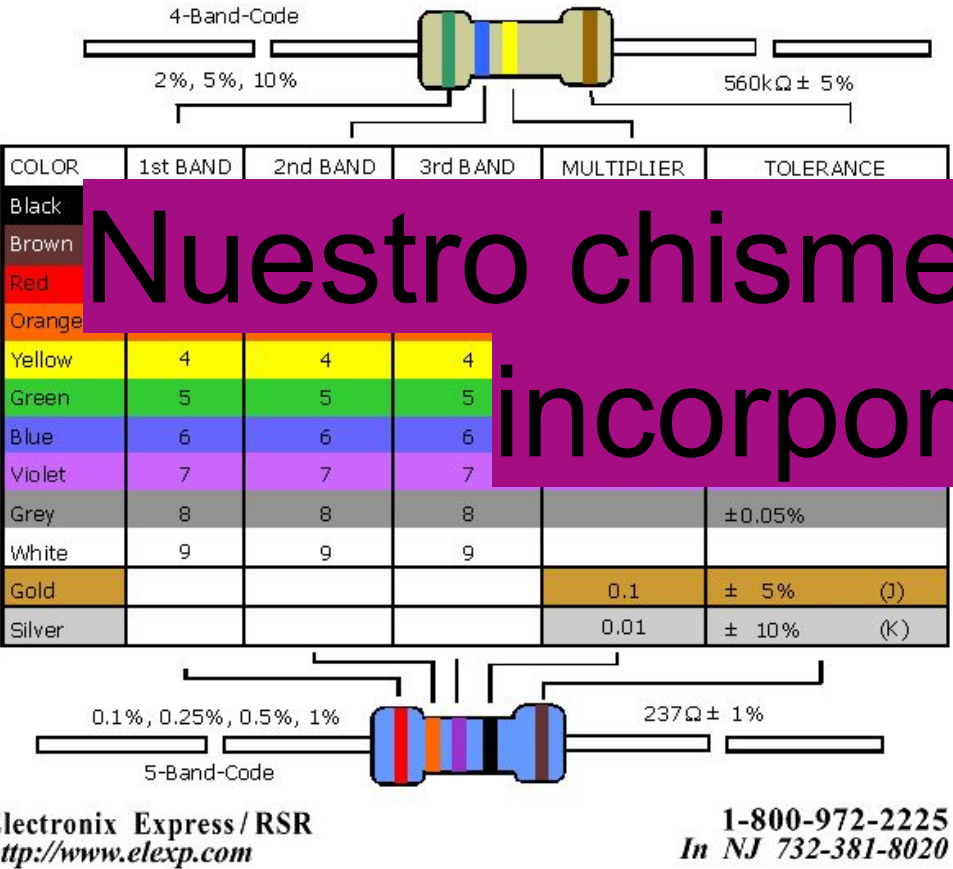
The most important parts on the Arduino board high lighted in red:

- 1: USB connector
- 2: Power connector
- 3: Automatic power switch
- 4: Digital pins
- 5: Analog pins
- 6: Power pins
- 7: Reset switch



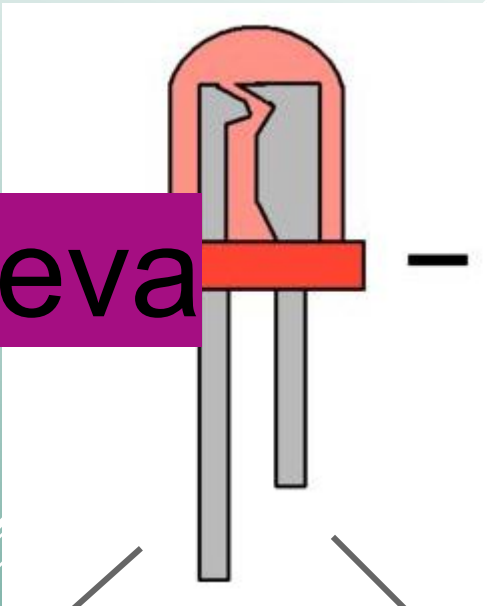
CARACTERÍSTICAS COMPONENTES

RESISTENCIA



Nuestro chisme ya los lleva incorporados

LED



ÁNODO

CÁTODO

CARACTERÍSTICAS COMPONENTES

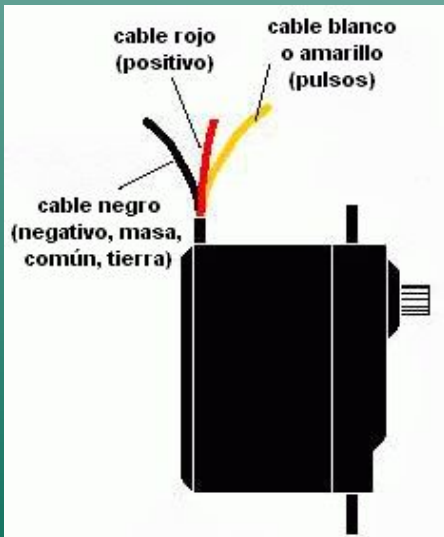
SERVO-MOTORES

180 GRADOS

CONTINUOS

Pines dig
la señal

Solo se les indica
ángulo y giran has
llegar a esa posición



Esto tampoco lo usamos,
quien usa cables aquí

- de 0 a 95: anti horario
- de 100 a 180: horario

PULSO

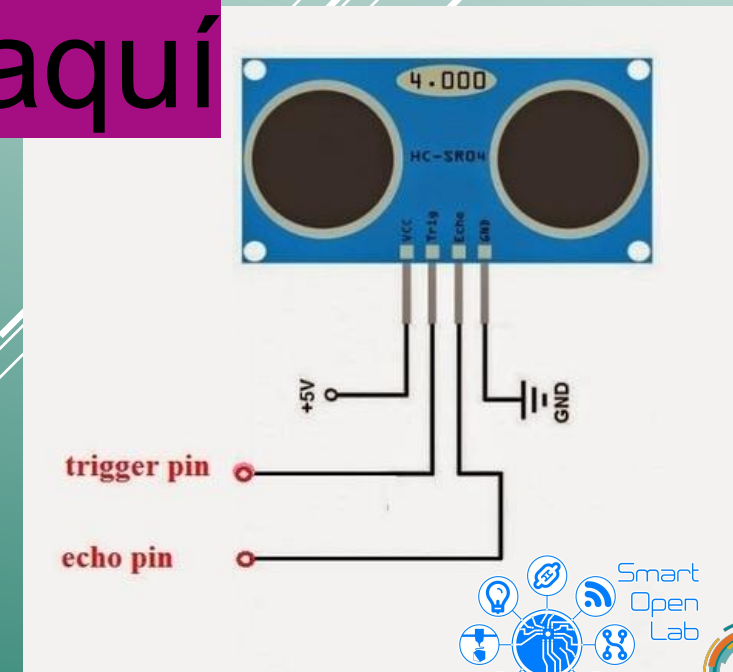
En función de la duración del pulso:

- de 0.9 a 1.4: horario
- 1.5= neutral
- de 1.9 a 2.2: anti horario

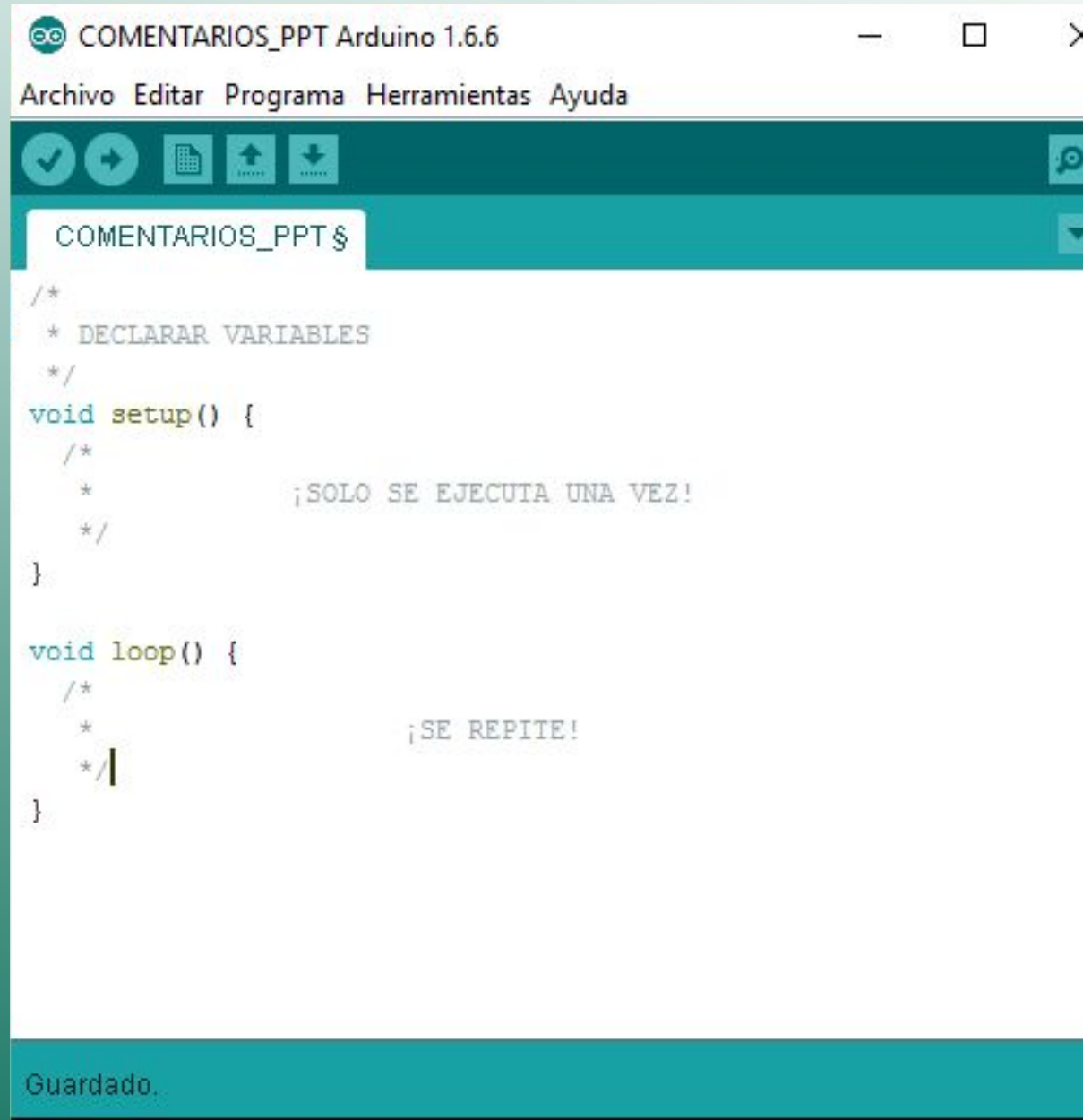
ULTRASONIDOS

Pines digitales PWM

de 1 (cm) a 4 (m)



ESTRUCTURA IDE ARDUINO



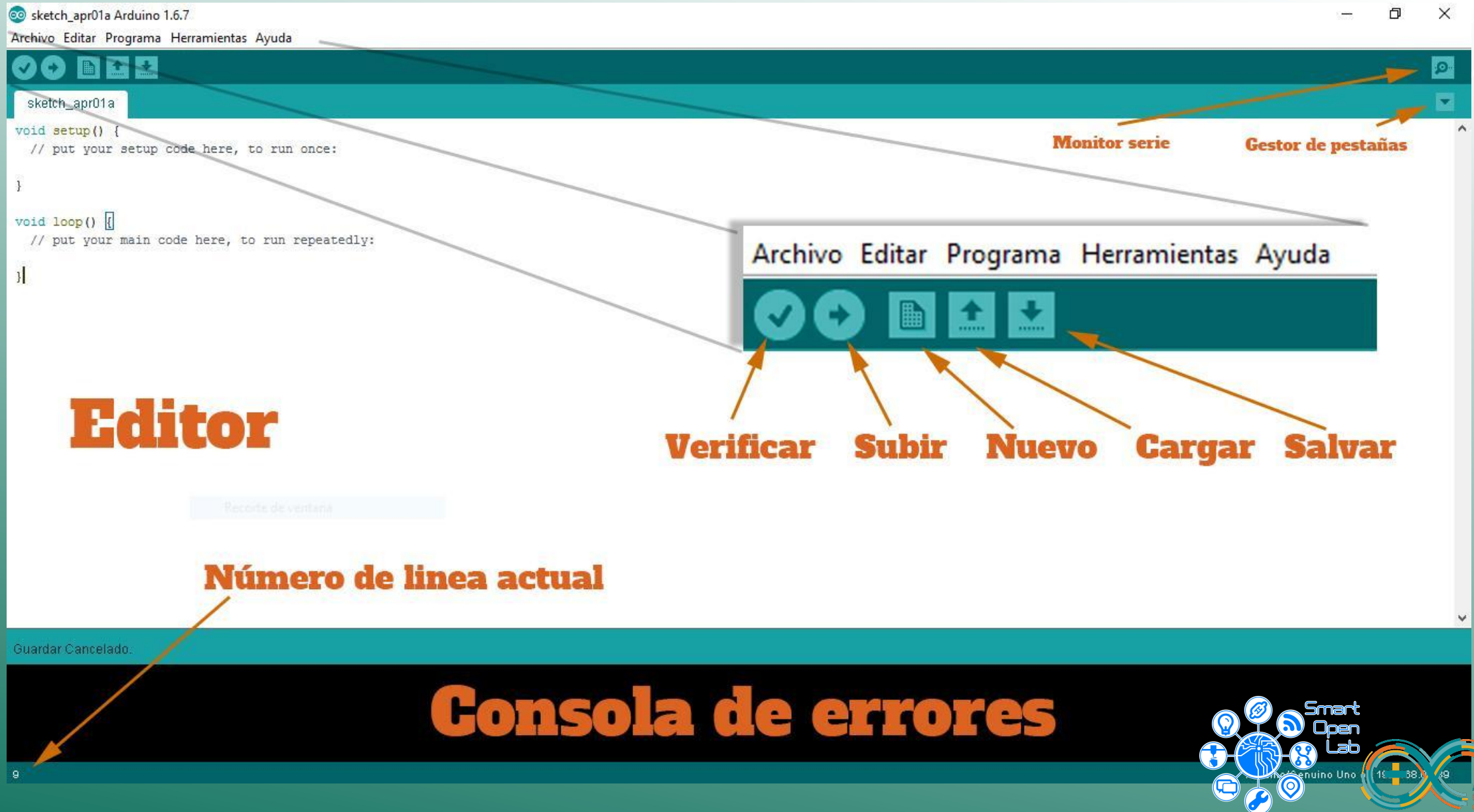
The screenshot displays the Arduino IDE 1.6.6 window. The title bar reads "COMENTARIOS_PPT Arduino 1.6.6". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for checking, running, opening a file, uploading, and downloading. The file name "COMENTARIOS_PPT \$" is shown in the top bar. The main editor area contains the following C++ code:

```
/*
 * DECLARAR VARIABLES
 */
void setup() {
  /*
   *           ¡SOLO SE EJECUTA UNA VEZ!
   */
}

void loop() {
  /*
   *           ¡SE REPITE!
   */
}
```

The status bar at the bottom indicates "Guardado..." (Saved...).

ESTRUCTURA IDE ARDUINO





Smart
Open
Lab

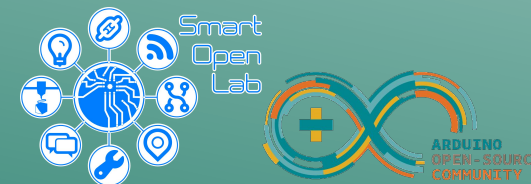


smartopenlab@gmail.com
[@solEpcc](mailto:solepcc.unex.es)

**PABELLÓN TELECOMUNICACIONES
(PLANTA BAJA)**

AHORA ES TU TURNO:

¡COMIENZA A CREAR!



Un led se enciende o no, bueno es posible

```
//LEDs rojo y azul
```

```
#define LED1 13
```

```
#define LED2 12
```

El setup hay que iniciar los pines

```
pinMode (LED1, OUTPUT);
```

```
pinMode (LED2, OUTPUT);
```

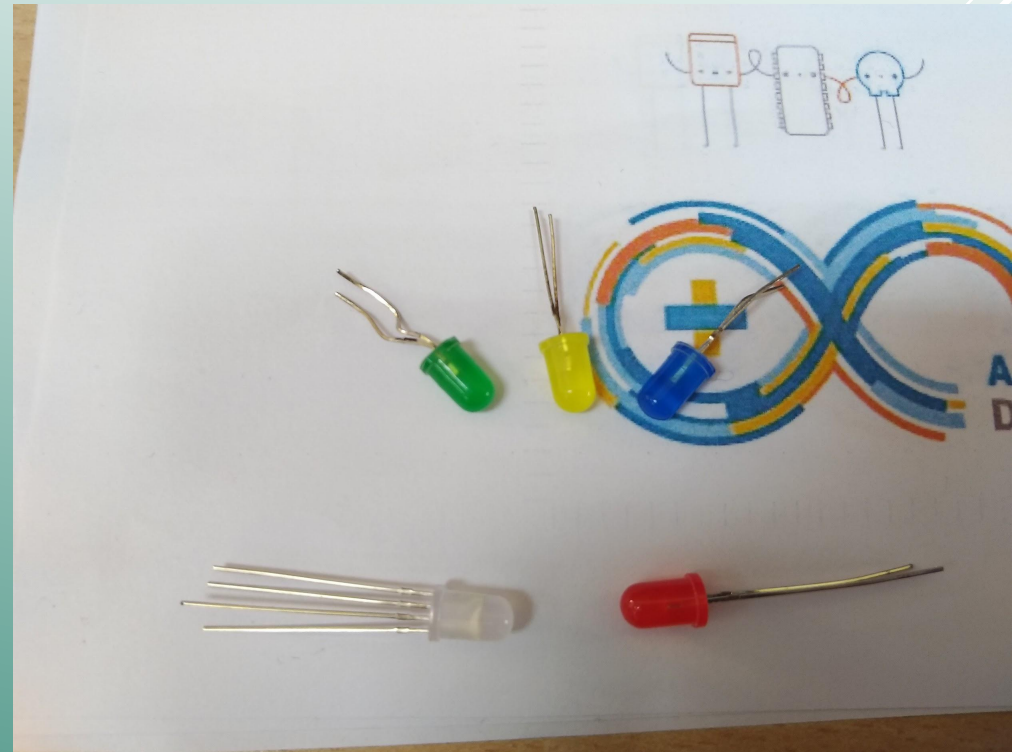
El loop a dar vueltas hasta el infinito

```
digitalWrite(LED2, HIGH);
```

```
delay(1000);
```

```
digitalWrite(LED2, LOW);
```

```
delay(500)
```



Un pequeño click para el hombre, un gran botón para la humanidad

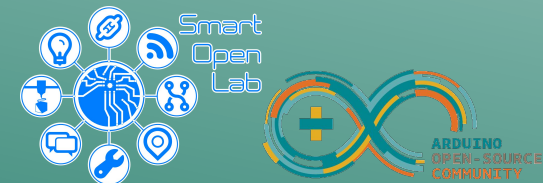
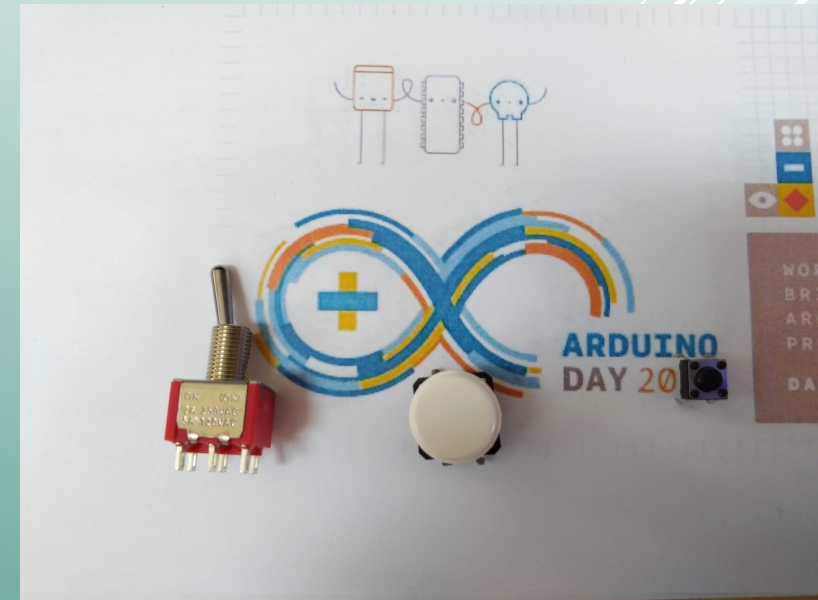
```
//Botones 1 y 2  
#define SW1 2  
#define SW2 3
```

El setup hay que iniciar los pines

```
//Para que funcionen en modo normal,  
si INPUT 0 sería encendido y 1 apagado  
pinMode (SW1, INPUT_PULLUP);  
pinMode (SW2, INPUT_PULLUP);
```

El loop a dar vueltas hasta el infinito

```
Serial.begin(9600);  
if (digitalRead(SW1))  
  Serial.println("¡Estoy encendido :-) !");  
else  
  Serial.println("Estoy apagado :-(");
```



La potencia sin control no sirve de nada, bueno solo si son DJs

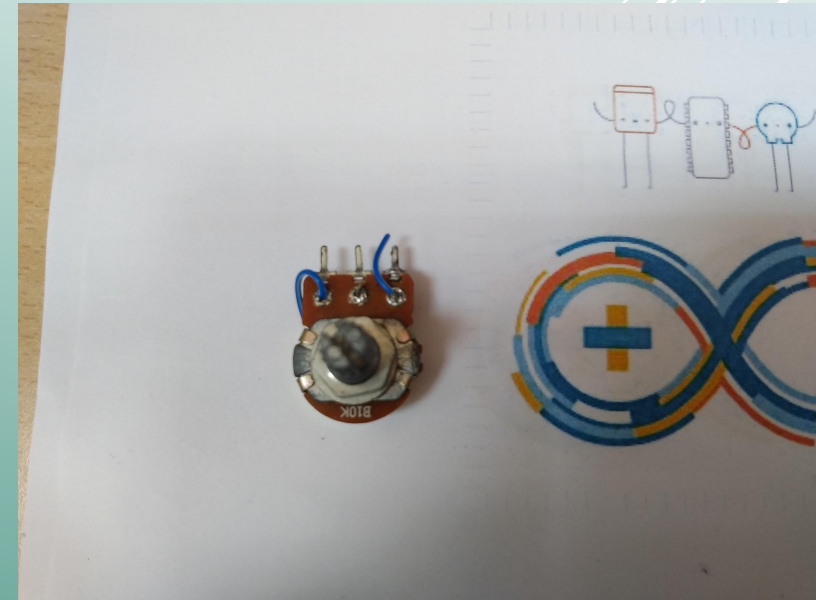
```
//Potenciometro  
#define POT A0  
int potReading;
```

El setup hay que iniciar los pines

```
Serial.begin(9600);  
pinMode (POT, INPUT);
```

El loop a dar vueltas hasta el infinito

```
Serial.print("Valor del potenciómetro: ");  
potReading = analogRead(POT);  
Serial.println(potReading);  
if (potReading > 1000)  
    Serial.println("Soy el potenciómetro y estoy a tope!");
```



Las sombras necesitan la luz para existir

```
//Sensor de luz, LDR  
#define LDR A1  
int LDRReading;
```

El setup hay que iniciar los pines

```
Serial.begin(9600);  
pinMode (LDR, INPUT);
```

El loop a dar vueltas hasta el infinito

```
Serial.print("Valor del LDR: ");  
LDRReading = analogRead(LDR);  
Serial.println(LDRReading);
```



Las sombras necesitan la luz para existir

```
//Buzzer, motor vibrador  
#define BUZZ 5
```

El setup hay que iniciar los pines
`pinMode (BUZZ, OUTPUT);`

El loop a dar vueltas hasta el infinito
`playMelody();`

```
void playMelody(){  
  int numTones = 10;  
  int tones[ ] = {261, 277, 294, 311, 330, 349,  
370, 392, 415, 440, 466, 494};  
  // mid C C# D D# E F F# G G# A  
  
  for (int i = 0; i < numTones; i++) {  
    tone(BUZZ, tones[i]);  
    delay(500);  
  }  
  noTone(BUZZ);  
}
```

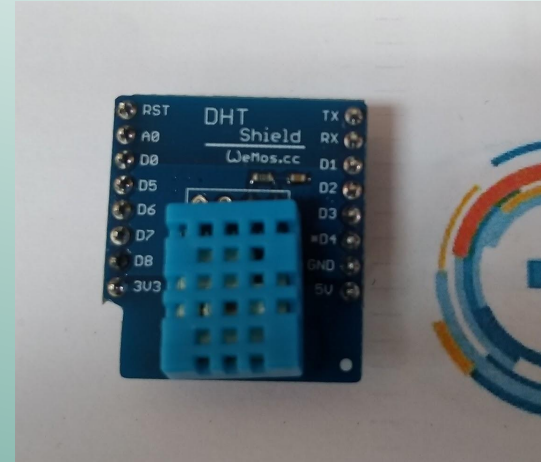


Para este no hay chiste, solo mide humedad y temperatura

```
//Sensor de temperatura y humedad
#define LM35 A2 //ACCURATE TEMP
#define DHT11 4 //HUM_TEMP
#include <SimpleDHT.h>
SimpleDHT11 TempHumSensor;
byte temperature = 0;
byte humidity = 0;
```

El setup hay que iniciar los pines

```
Serial.begin(9600);
pinMode (LM35, INPUT);
pinMode (DHT11, INPUT);
```



El loop a dar vueltas hasta el infinito

```
int err = SimpleDHTErrSuccess;
if ((err = TempHumSensor.read(DHT11,
&temperature, &humidity, NULL)) ==
SimpleDHTErrSuccess) {
    Serial.print((int)temperature);
    Serial.print("°C, ");
    Serial.print((int)humidity);
    Serial.println("% Humedad");
}
```