

Introducción a la Ingeniería

ALGORITMOS BÁSICOS

Presentación No. 2

ALGORITMOS BÁSICOS

OBJETIVOS

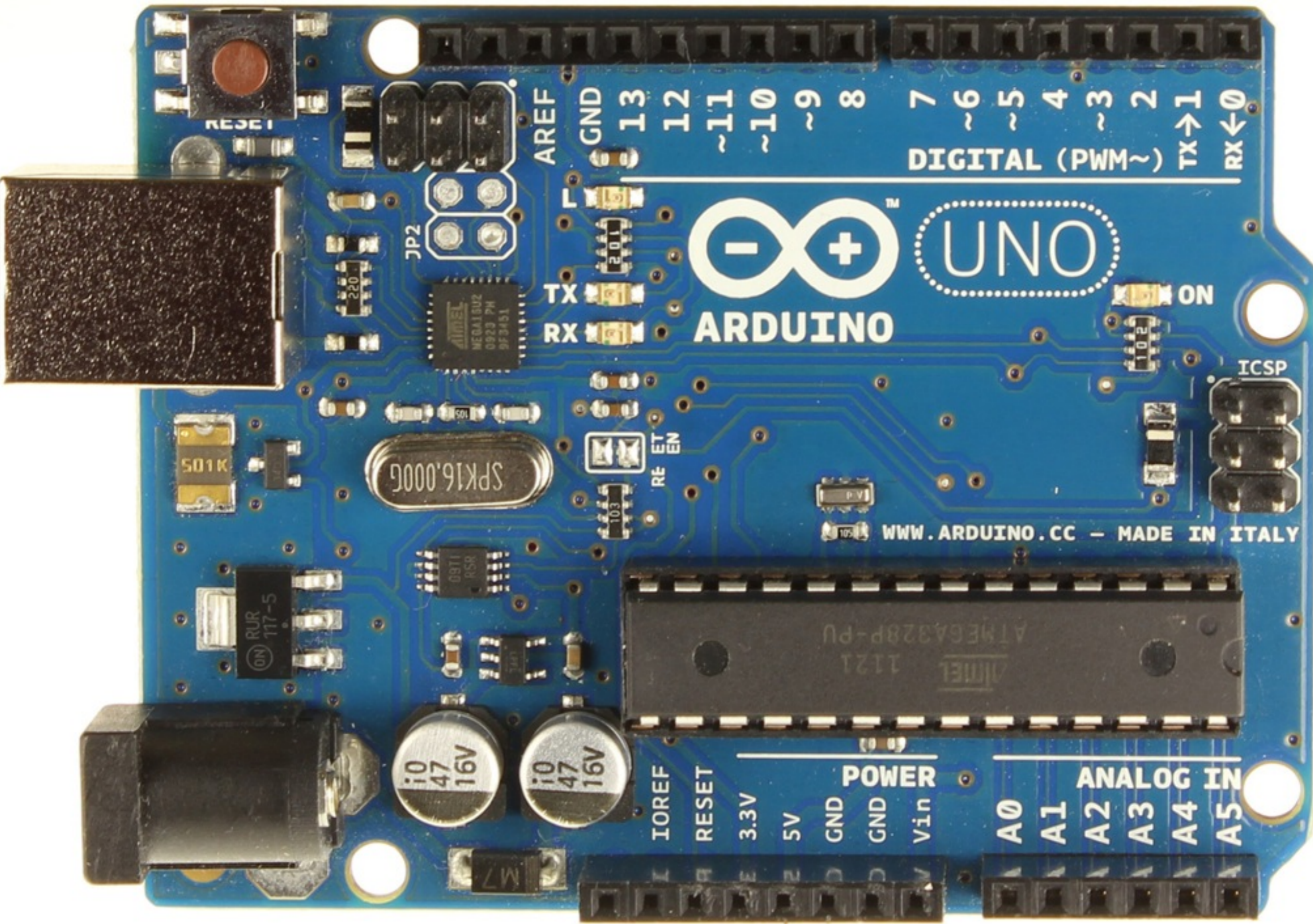
- Entender y utilizar los ALGORITMOS
- Concebir, proponer, implementar y evaluar un pequeño PROYECTO
- Empezar a distinguir entre señales CONTINUAS y DISCRETAS



EL ARDUINO

VISITAR:

- ❖ <http://arduino.cc>
- ❖ <http://www.instructables.com/id/Arduino-Projects/>
- ❖ <http://playground.arduino.cc/projects/ideas>
- ❖ <http://codeluino.com/projects/>
- ❖ <http://fritzing.org/projects/>



ARDUINO UNO

WWW.ARDUINO.CC - MADE IN ITALY

AREF GND 13 12 11 10 9 8 7 6 5 4 3 2 1 0
DIGITAL (PWM ~) TX ← RX →

IOREF RESET 3.3V 5V GND GND Vin
A0 A1 A2 A3 A4 A5
ANALOG IN

ATMEL MEGA16U2 0923 PH 9F3451

ATMEL ATMEGA328P-PU 1121

SPK16.000G

i0 47 16V

i0 47 16V

501K

RUR 117-5

RESET

JP2

TX RX

RE ET EN

ICSP

ON

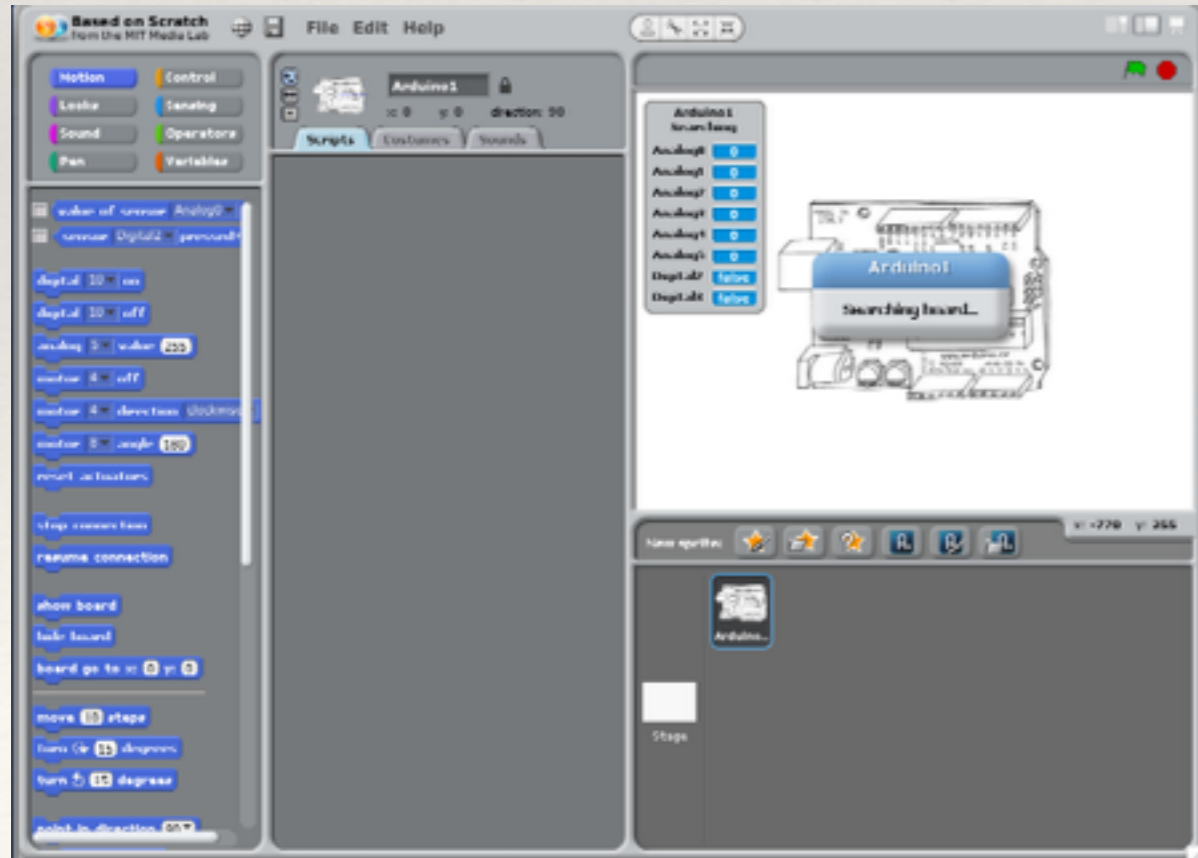
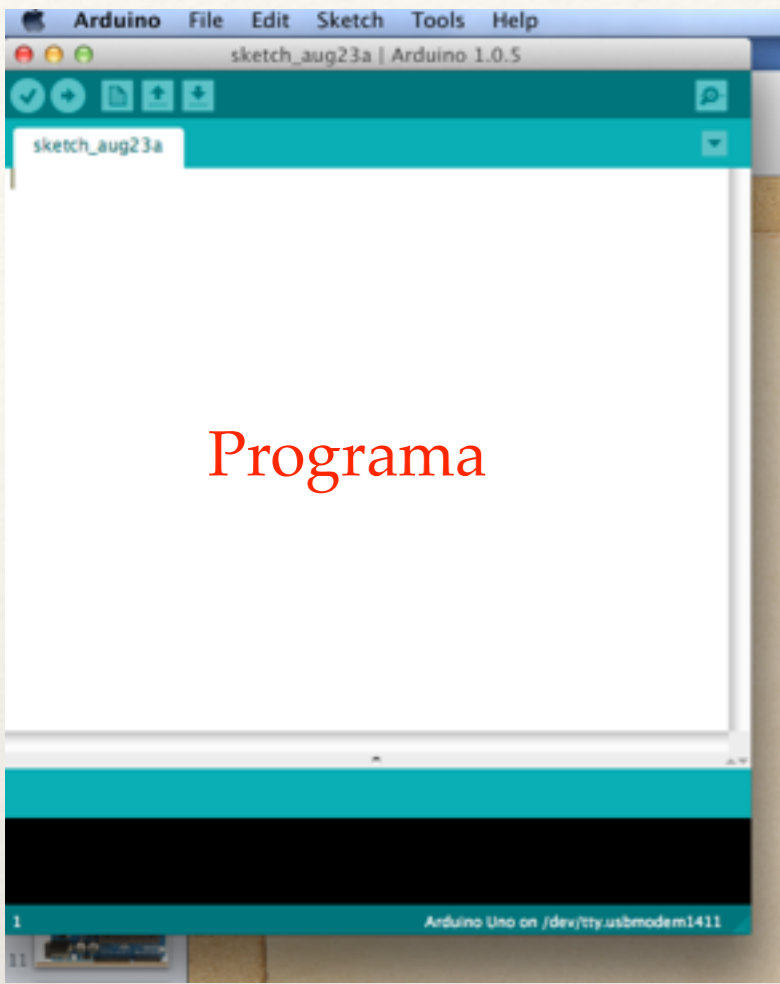
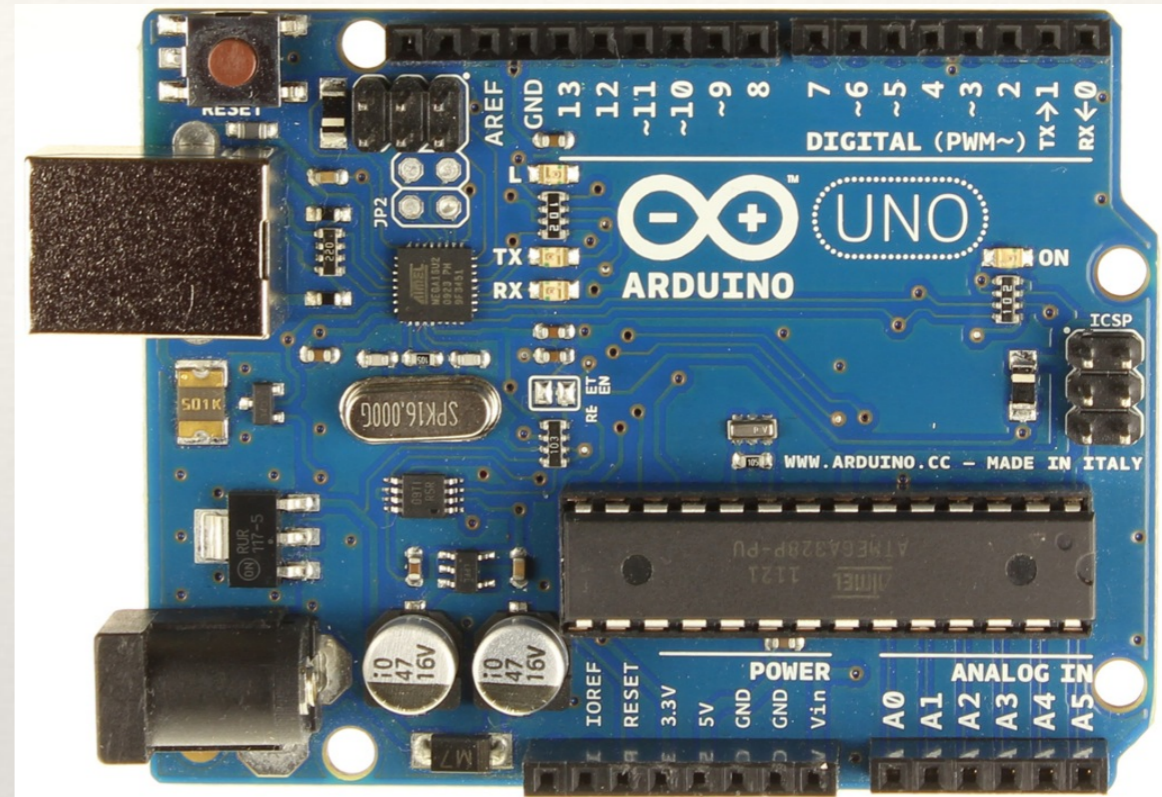
M7

Entorno de desarrollo con ARDUINO

IDE de Arduino

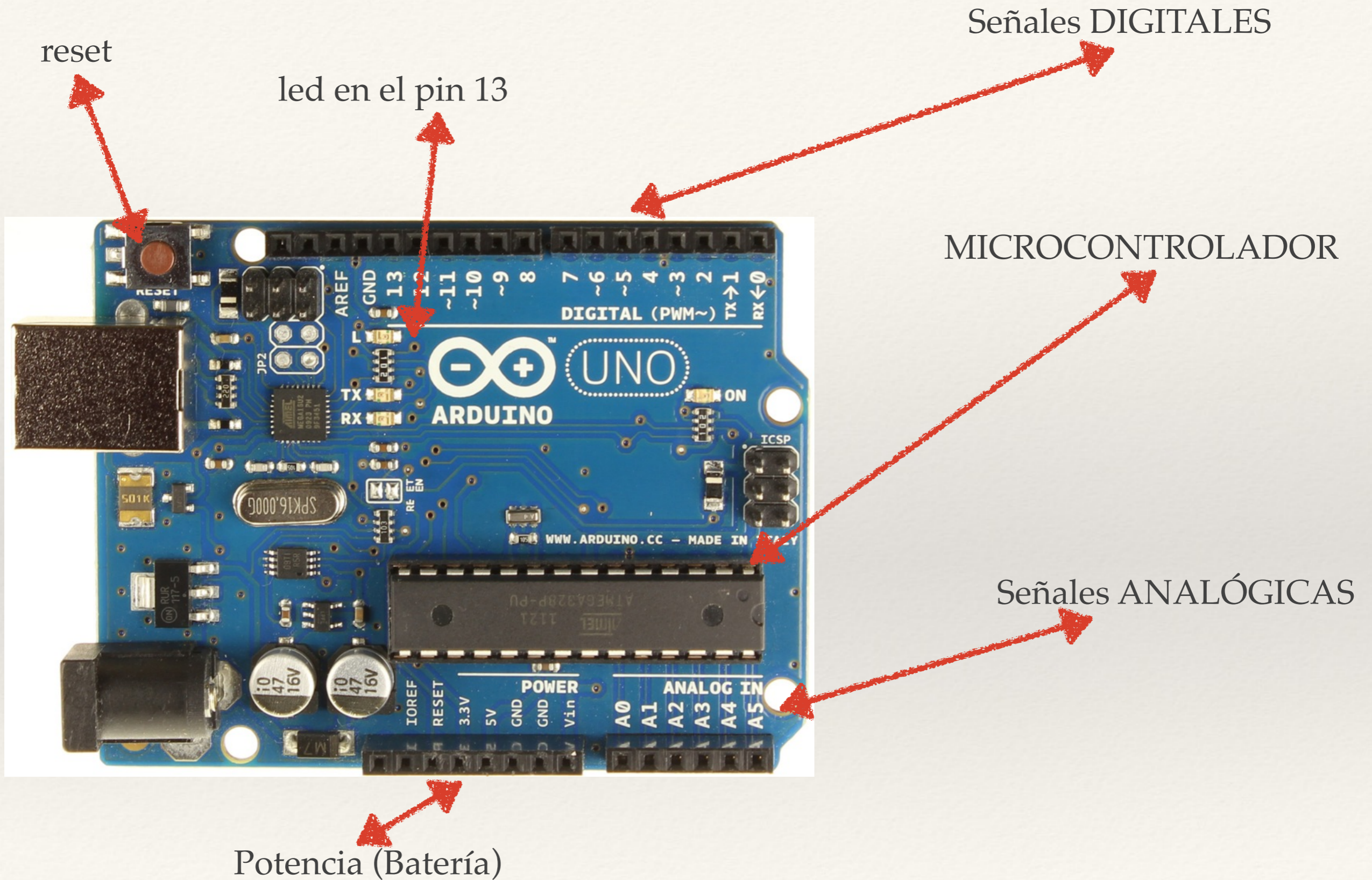
Programa

Cable USB



S4A

Elementos importantes del ARDUINO



SISTEMA



Ejemplo: SISTEMA DE CONTROL DE TEMPERATURA

- ENTRADAS: Medida de temperatura, valor deseado de T°
- SALIDAS: "Actuador" sobre el motor del enfriador, "display"
- SISTEMA : Controlador

Primer programa en ARDUINO

```
Blink | Arduino 1.0.5
Blink §
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

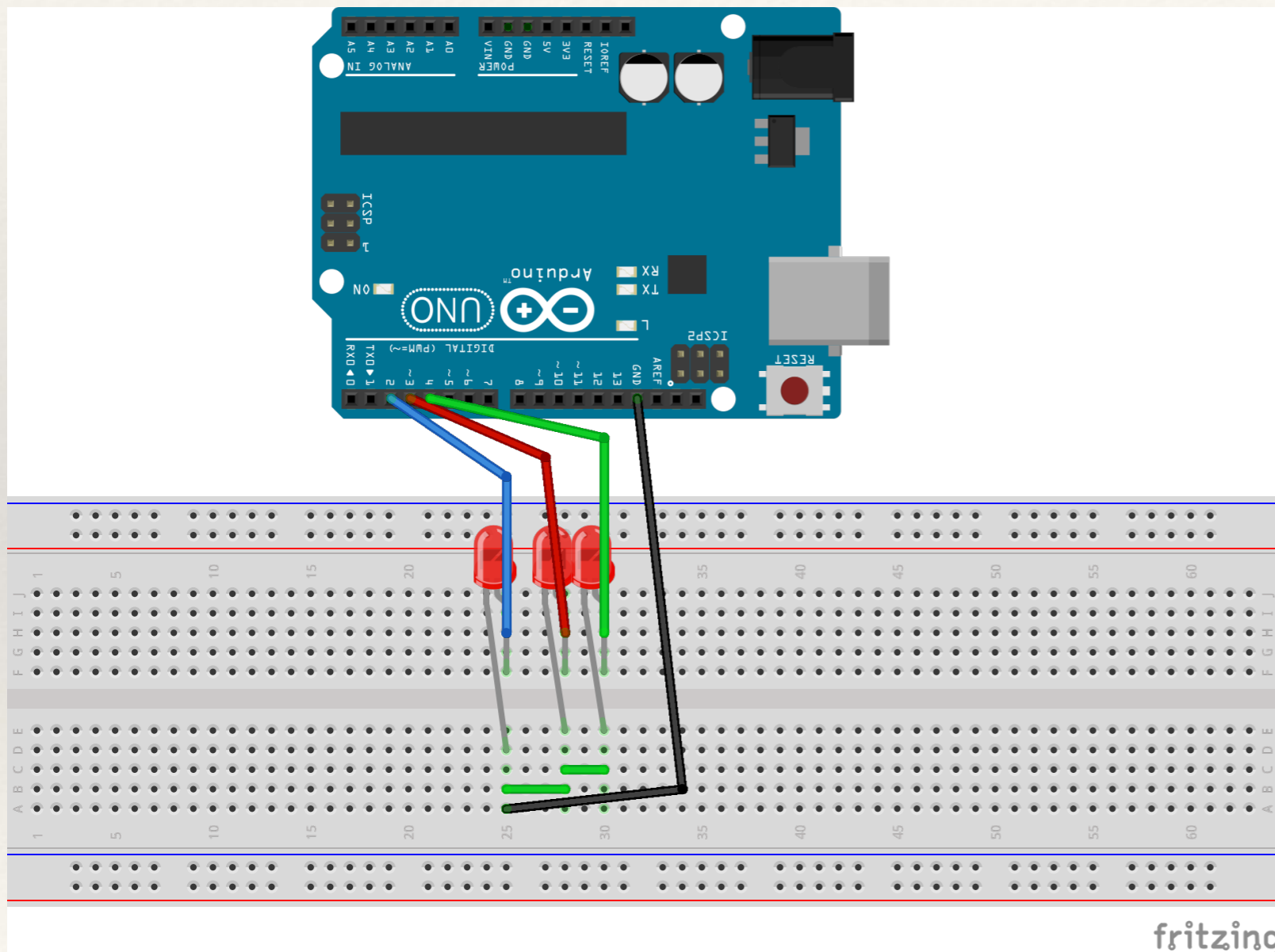
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

// the loop routine runs over and over again forever:
void loop() {
}
```

Prog1

Segundo programa

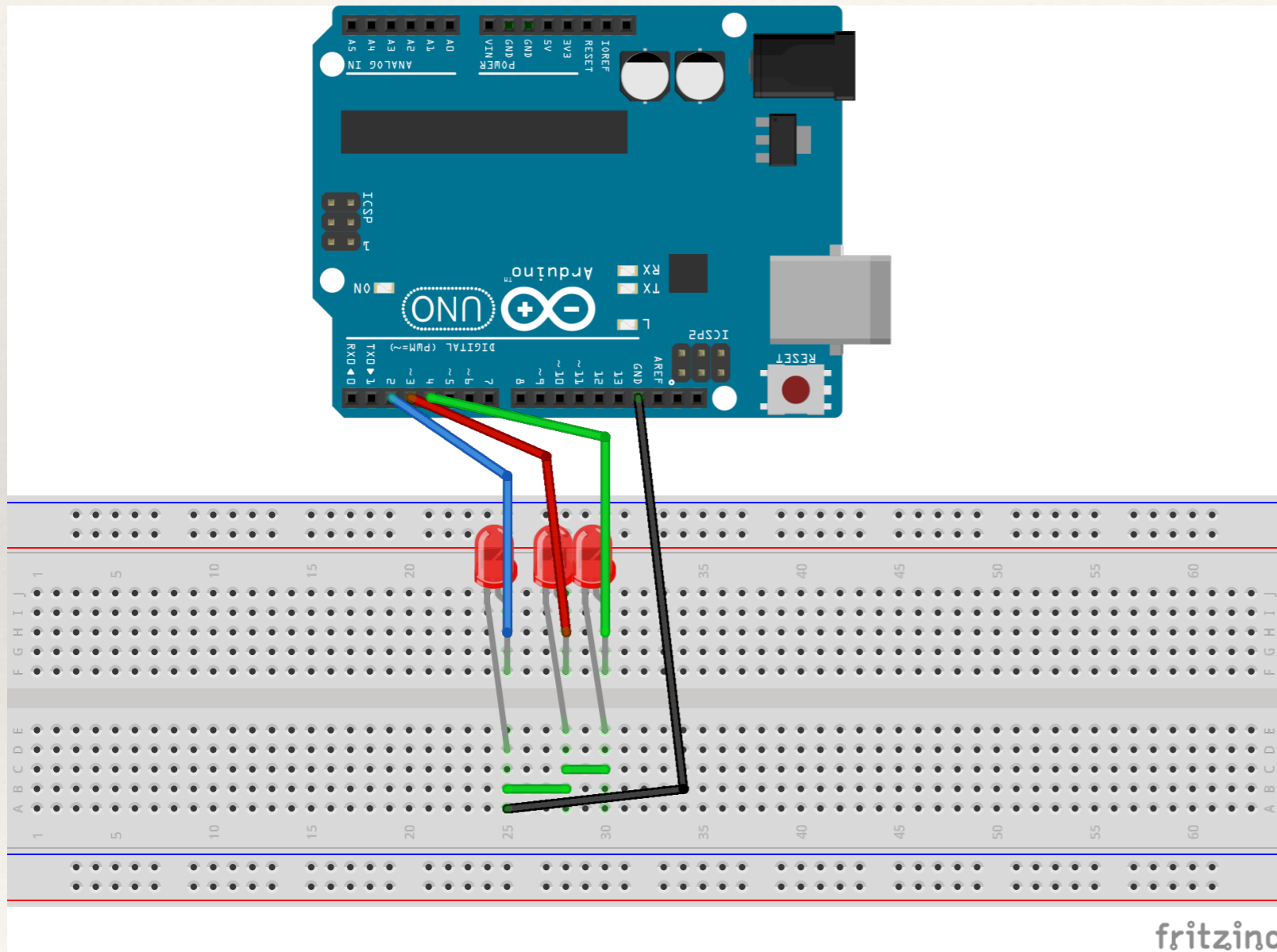
Prog2



El programa debe:
Encender led 2
esperar 1 segundo
apagar led 2
Encender led 3
esperar 1 segundo
apagar led 3
Encender led 4
esperar 1 segundo
Apagar led 4

Modificación

Prog2A

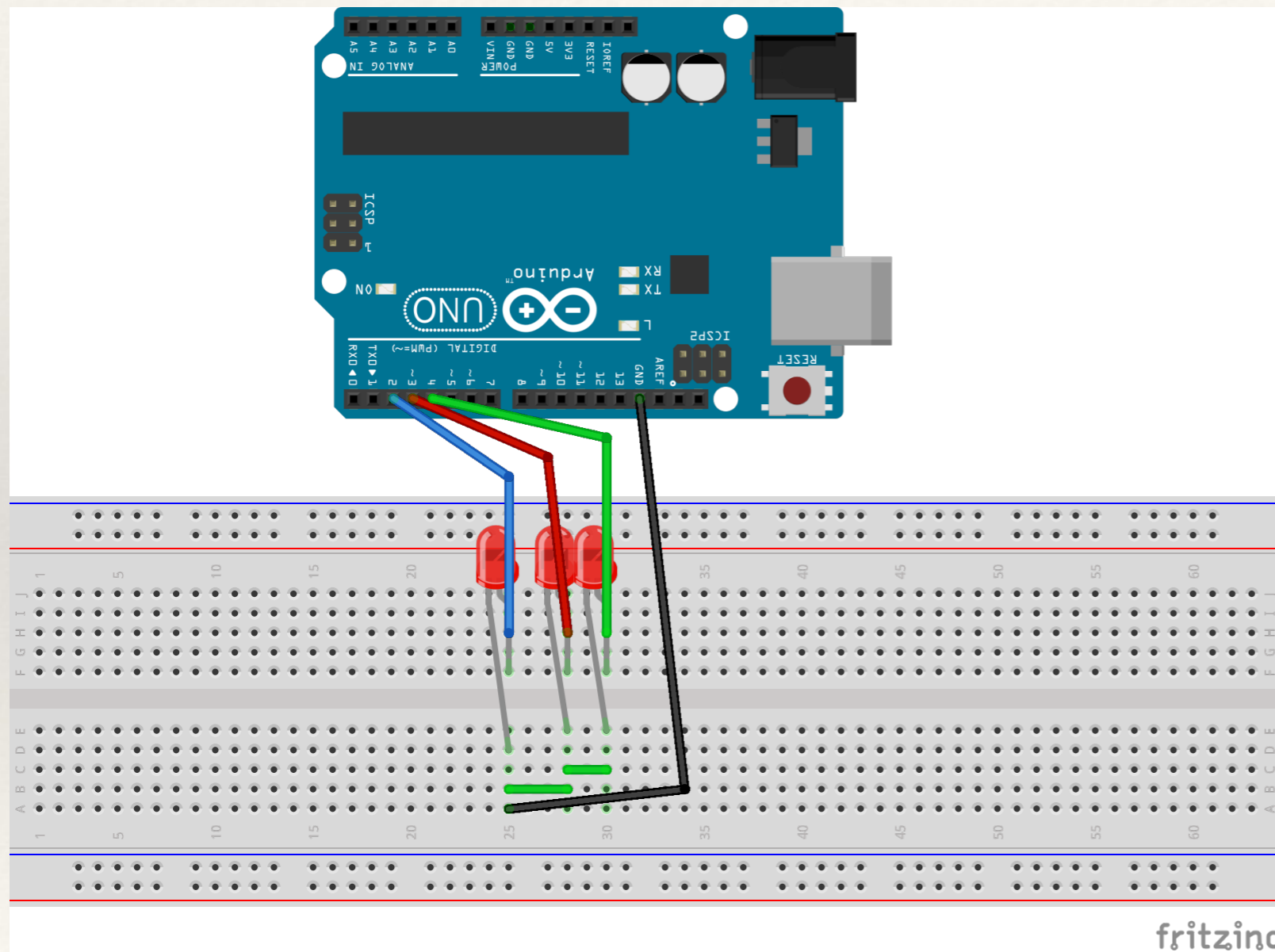


El programa debe:

Encender led 2
esperar 1 segundo
Encender led 3
esperar 1 segundo
Encender led 4
esperar 1 segundo
Apagar led 4
esperar 1 segundo
Apagar led 3
esperar 1 segundo
Apagar led 2
esperar 1 segundo

Tercer programa

Se requiere
repetir la
secuencia
5 veces



El programa debe:
Encender led 2
esperar 1 segundo
apagar led 2
Encender led 3
esperar 1 segundo
apagar led 3
Encender led 4
esperar 1 segundo
Apagar led 4



BareMinimum §

```
int led1 = 2;
int led2 = 3;
int led3 = 4;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  digitalWrite(led1, HIGH);
  delay(1000);
  digitalWrite(led1, LOW);
  delay(1000);
  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(1000);
  digitalWrite(led3, HIGH);
  delay(1000);
  digitalWrite(led3, LOW);
  delay(1000);
}

void loop() {
  // put your main code here, to run repeatedly
```

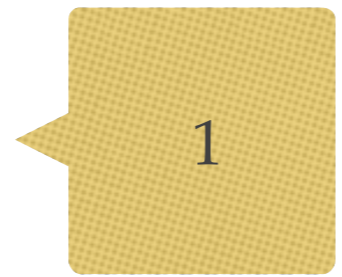
Copio y pego 5 veces!!!!



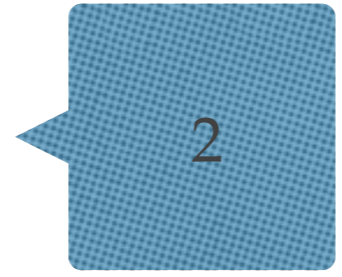
BareMinimum §

```
int led1 = 2;  
int led2 = 3;  
int led3 = 4;  
void setup() {  
  pinMode(led1,OUTPUT);  
  pinMode(led2,OUTPUT);  
  pinMode(led3,OUTPUT);  
  digitalWrite(led1,HIGH);  
  delay(1000);  
  digitalWrite(led1,LOW);  
  delay(1000);  
  digitalWrite(led2,HIGH);  
  delay(1000);  
  digitalWrite(led2,LOW);  
  delay(1000);  
  digitalWrite(led3,HIGH);  
  delay(1000);  
  digitalWrite(led3,LOW);  
  delay(1000);  
  delay(1000);  
  digitalWrite(led1,LOW);  
  delay(1000);  
  digitalWrite(led2,HIGH);  
  delay(1000);  
  digitalWrite(led2,LOW);  
  delay(1000);  
  digitalWrite(led3,HIGH);  
  delay(1000);  
  digitalWrite(led3,LOW);  
  delay(1000);  
  delay(1000);  
  digitalWrite(led1,LOW);  
  delay(1000);  
  digitalWrite(led2,HIGH);  
  delay(1000);  
  digitalWrite(led2,LOW);  
  delay(1000);  
}
```

digitalWrite(led1,HIGH);
delay(1000);
digitalWrite(led1,LOW);
delay(1000);
digitalWrite(led2,HIGH);
delay(1000);
digitalWrite(led2,LOW);
delay(1000);
digitalWrite(led3,HIGH);
delay(1000);
digitalWrite(led3,LOW);
delay(1000);



delay(1000);
digitalWrite(led1,LOW);
delay(1000);
digitalWrite(led2,HIGH);
delay(1000);
digitalWrite(led2,LOW);
delay(1000);
digitalWrite(led3,HIGH);
delay(1000);
digitalWrite(led3,LOW);
delay(1000);



delay(1000);
digitalWrite(led1,LOW);
delay(1000);
digitalWrite(led2,HIGH);
delay(1000);
digitalWrite(led2,LOW);
delay(1000);



Y si quisiéramos repetir 100 veces????



BareMinimum §

```
int led3 = 4;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
```

```
i=0;
```

```
while(i<5);
```

```
  digitalWrite(led1, HIGH);
  delay(1000);
  digitalWrite(led1, LOW);
  delay(1000);
  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(1000);
  digitalWrite(led3, HIGH);
  delay(1000);
  digitalWrite(led3, LOW);
  delay(1000);
```

```
  i=i+1
```

```
}
```

i ← CONTADOR

ARDUINO

[Buy](#) [Download](#) [Getting Started](#) [Learning](#) [Reference](#) [Hardware](#) [FAQ](#)

[Reference](#) [Language](#) [Libraries](#) [Comparison](#) [Changes](#)

while loops

Description

while loops will loop continuously, and infinitely, until the expression inside the parenthesis, () becomes false. Something must change the tested variable, or the **while** loop will never exit. This could be in your code, such as an incremented variable, or an external condition, such as testing a sensor.

Syntax

```
while(expression){
  // statement(s)
}
```

Parameters

expression - a (boolean) C statement that evaluates to true or false

Example

```
var = 0;
while(var < 200){
  // do something repetitive 200 times
  var++;
}
```



```
BareMinimum 5
int led1 = 2;
int led2 = 3;
int led3 = 4;
int i;
void setup() {
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  i=0;
  while(i<5);
  prende();
  i=i+1;
}
}
void loop() {
}
void prende(){
  digitalWrite(led1,HIGH);
  delay(1000);
3  digitalWrite(led1,LOW);
  delay(1000);
  digitalWrite(led2,HIGH);
  delay(1000);
  digitalWrite(led2,LOW);
  delay(1000);
4  digitalWrite(led3,HIGH);
  delay(1000);
  digitalWrite(led3,LOW);
  delay(1000);
}
```

INVOCACIÓN DE LA
FUNCIÓN

FUNCIÓN

Más ELEGANTE!!!!!!

```

Secuencia2 §
int led1 = 2;
int led2 = 3;
int led3 = 4;
int i;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);

  for(i=0; i<5; i++) {
    prنده();
    i=i+1;
  }
}
void loop() {
}
void prنده(){
  digitalWrite(led1, HIGH);
  delay(1000);
  digitalWrite(led1, LOW);
  delay(1000);
  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led2, LOW);
  delay(1000);
  digitalWrite(led3, HIGH);
  delay(1000);
  digitalWrite(led3, LOW);
  delay(1000);
}

```

for statements

Description

The **for** statement is used to repeat a block of statements enclosed in curly braces. An increment counter is usually used to increment and terminate the loop. The **for** statement is useful for any repetitive operation, and is often used in combination with arrays to operate on collections of data/pins.

There are three parts to the **for** loop header:

```

for (initialization; condition; increment) {

//statement(s);

}

```

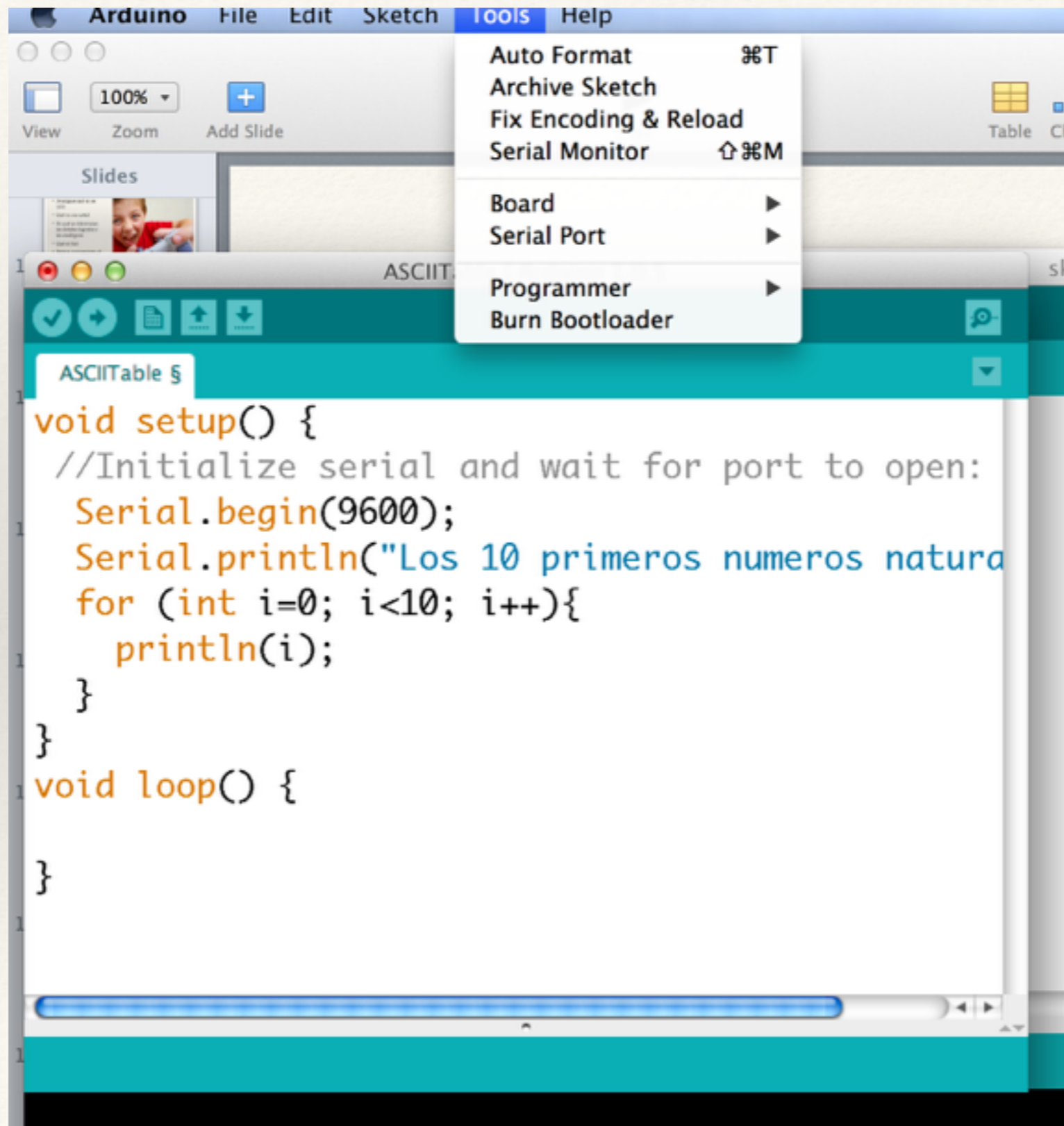
```

parenthesis
  declare variable (optional)
  initialize
  test
  increment or decrement
for(int x = 0; x < 100; x++) {
  println(x); // prints 0 to 99
}

```

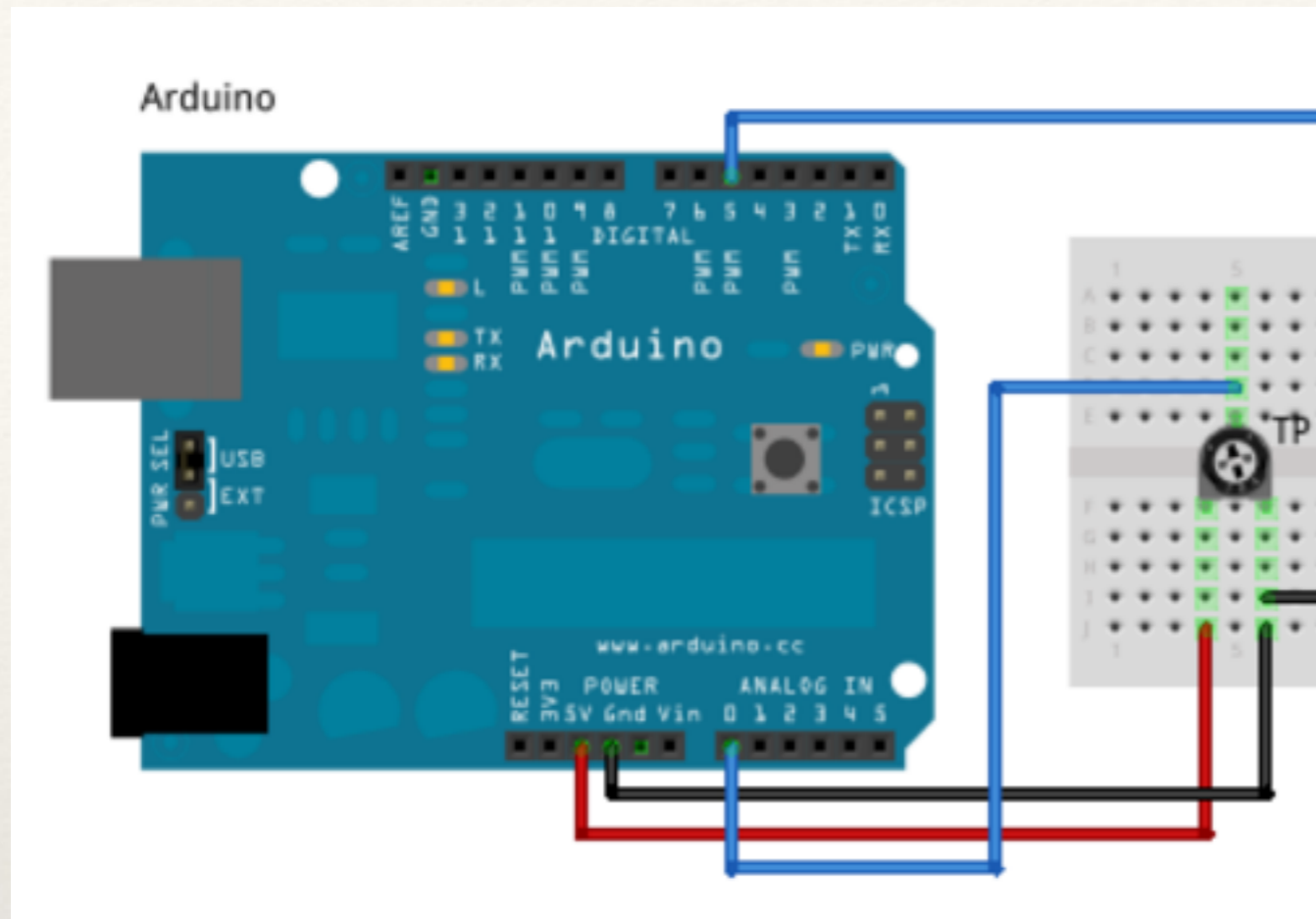
Mejor todavía!!!!!!!

El puerto serial



The image shows the Arduino IDE interface. The 'TOOLS' menu is open, displaying options: Auto Format (⌘T), Archive Sketch, Fix Encoding & Reload, Serial Monitor (⇧⌘M), Board, Serial Port, Programmer, and Burn Bootloader. Below the menu, the code editor shows the following C++ code:

```
void setup() {  
  //Initialize serial and wait for port to open:  
  Serial.begin(9600);  
  Serial.println("Los 10 primeros numeros naturales");  
  for (int i=0; i<10; i++){  
    println(i);  
  }  
}  
  
void loop() {  
  
}
```



Lectura de Potenciómetro Salida por Serial

Prog4

```
int analogPin = 0;
int val;
void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);

  // prints title with ending line break
  Serial.println("LECTURA DE POTENCIOMETRO");
}

void loop() {

  val = analogRead(analogPin); // read the input pin
  Serial.println(val);
  delay(400); // debug value
}
```

EJERCICIO

Prog5

```
Blink | Arduino 1.0.5
Blink 5
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

// the loop routine runs over and over again forever:
void loop() {
}
```

Se requiere repetir la secuencia 5 veces, de tal modo que el retardo entre el encendido y el apagado se incremente en cada repetición en un 10%

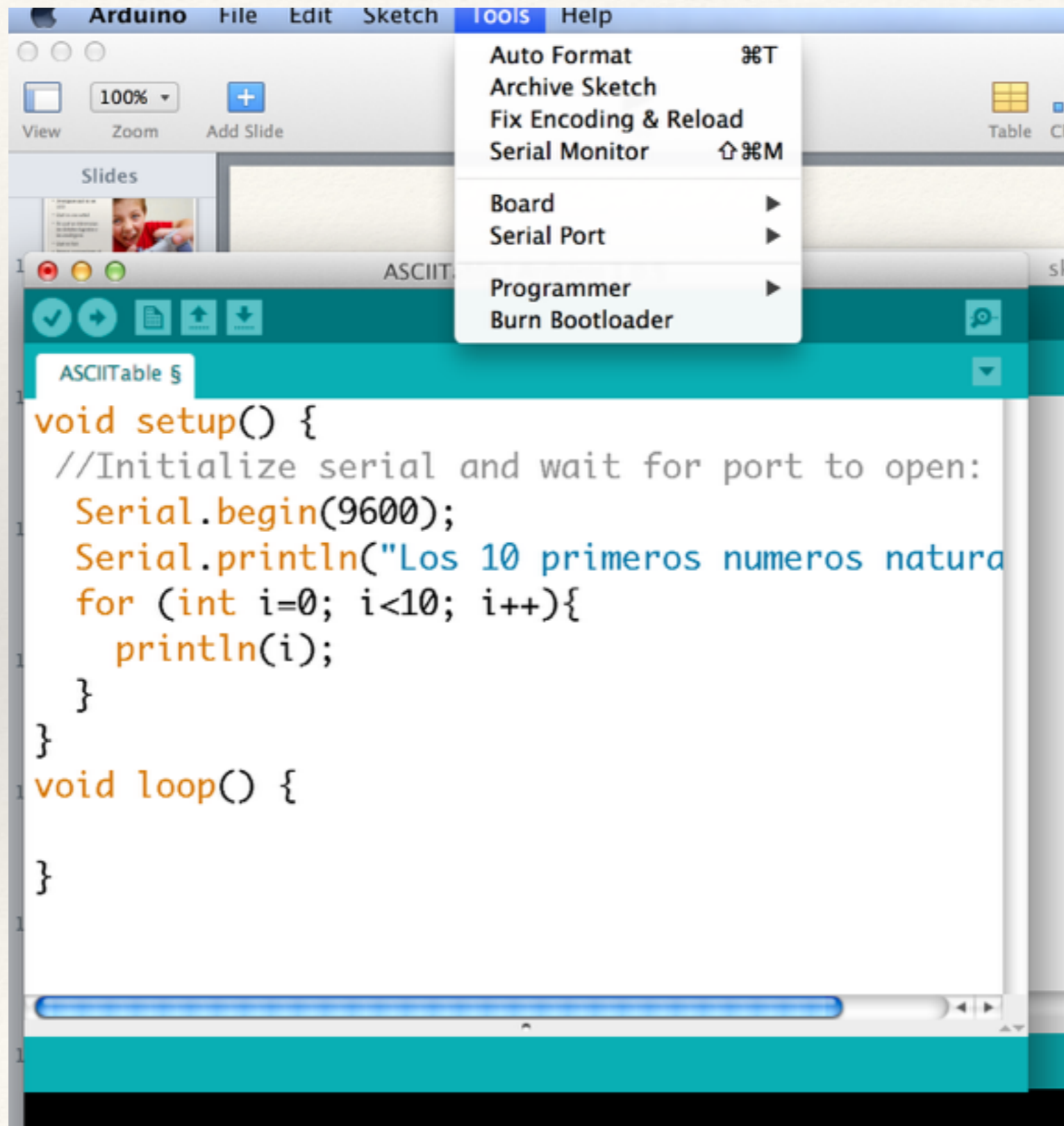


Prog6

Agregar una nueva secuencia a continuación de la anterior que repita 5 veces el encendido y apagado del led, de tal forma que el retardo entre el encendido y el apagado se vaya disminuyendo en un 10%

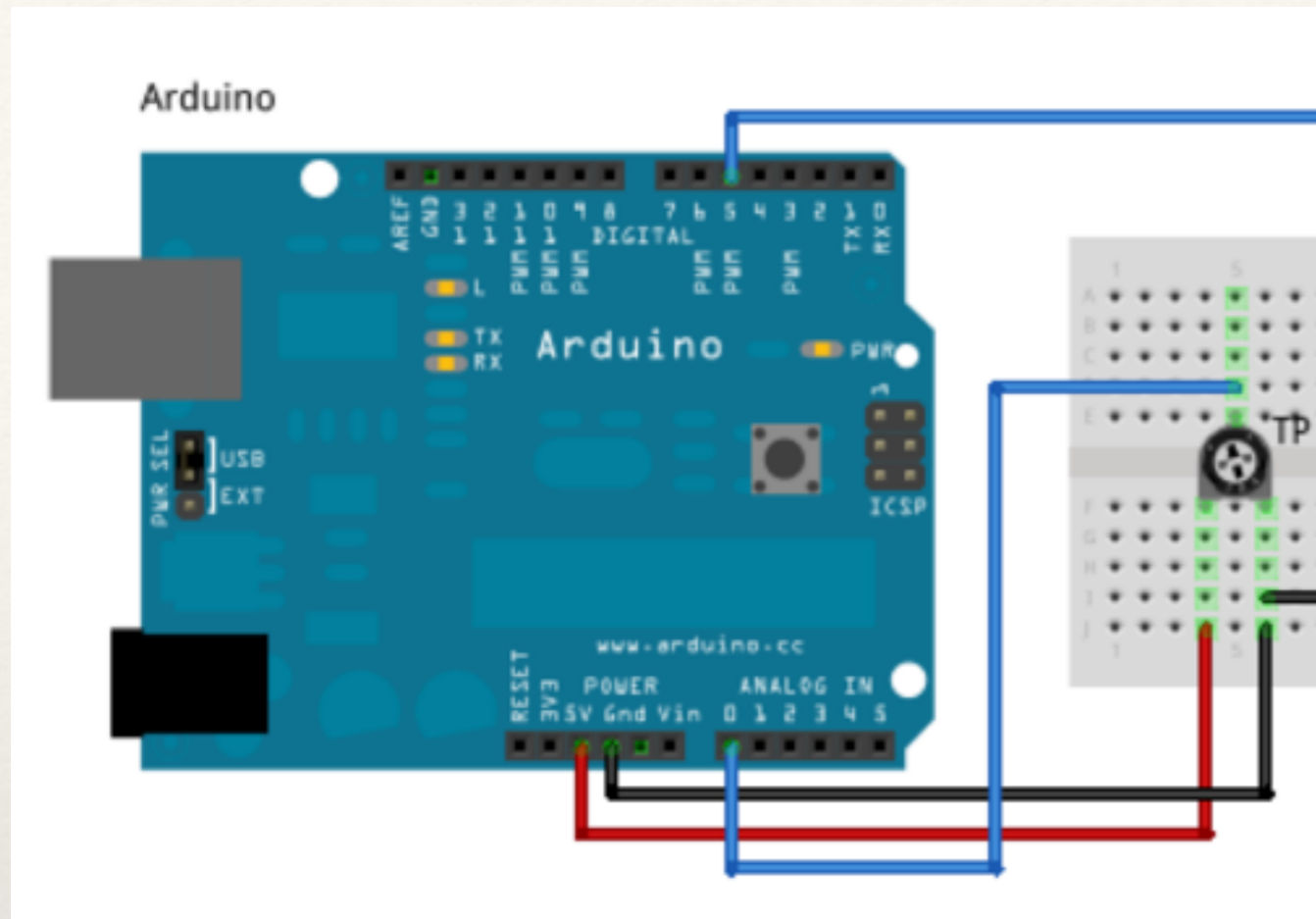
Agrégueme al programa que vaya mostrando por el PUERTO SERIAL el valor del retardo en cada ciclo

El puerto serial



The image shows the Arduino IDE interface. The 'TOOLS' menu is open, displaying options: Auto Format (⌘T), Archive Sketch, Fix Encoding & Reload, Serial Monitor (⇧⌘M), Board, Serial Port, Programmer, and Burn Bootloader. Below the menu, a code editor window titled 'ASCIITable §' contains the following C++ code:

```
1 void setup() {  
2   //Initialize serial and wait for port to open:  
3   Serial.begin(9600);  
4   Serial.println("Los 10 primeros numeros naturales");  
5   for (int i=0; i<10; i++){  
6     println(i);  
7   }  
8 }  
9 void loop() {  
10 }  
11
```



Lectura de Potenciómetro Salida por Serial

```
int analogPin = 0;
int val;
void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);

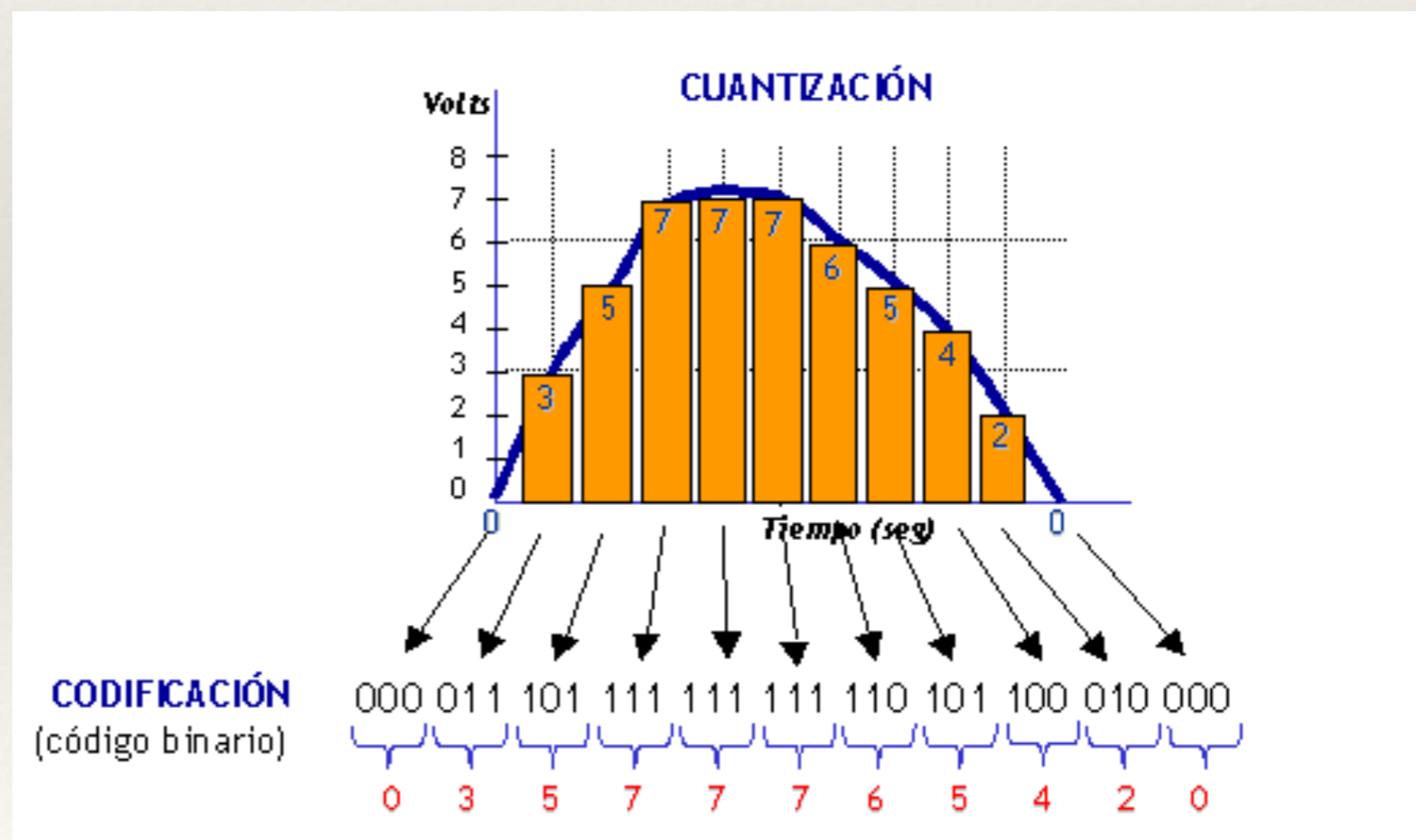
  // prints title with ending line break
  Serial.println("LECTURA DE POTENCIOMETRO");
}

void loop() {

  val = analogRead(analogPin); // read the input pin
  Serial.println(val);
  delay(400); // debug value
}
```




ADC





SISTEMA BINARIO

DECIMAL	BINARIO				
	8421	8	4	2	1
0	0000	0	0	0	0
1	0001	0	0	0	1
2	0010	0	0	1	0
3	0011	0	0	1	1
4	0100	0	1	0	0
5	0101	0	1	0	1
6	0110	0	1	1	0
7	0111	0	1	1	1
8	1000	1	0	0	0
9	1001	1	0	0	1
10	1010	1	0	1	0
11	1011	1	0	1	1
12	1100	1	1	0	0
13	1101	1	1	0	1
14	1110	1	1	1	0
15	1111	1	1	1	1

Tarea

Prog7

- 1) Realice el montaje mostrado en la figura y pruebe el programa dado. Reporte sus resultados.

```
/* www.dynamoelectronics.com
Este ejemplo enciende y apaga un led variando su intensidad, este led no puede ser el del pin13 debido
a su tipo, por lo tanto se recomienda usar un led difuso estandar.
La rutina consta de dos for, el primero aumenta los valores de la variable "a" y el segundo los disminuye
estos valores se asignan a la función analogWrite() */

int led = 9;          // Asignación del puerto 9 a la palabra led

void setup() {       // Rutina de condiciones iniciales

  pinMode(led, OUTPUT); // Declaración de led (pin9) como salida
}

void loop() {        // Rutina principal
  for (int a=0; a<255; a++) // se declara "a" como int
  { analogWrite(led, a);    // Se asigna el valor de a al puerto de salida
    delay(20);              // retardo de 20mseg
  }
  for ( a=255; a>1; a--)   // se declara "a" como int
  { analogWrite(led, a);    // Se asigna el valor de a al puerto de salida
    delay(20);              // retardo de 20mseg
  }
}
```



- 2) Investigue qué es y para que sirve PWM