

Introducción a la Ingeniería

ALGORITMOS BÁSICOS

Presentación No. 3

ALGORITMOS BÁSICOS

OBJETIVOS

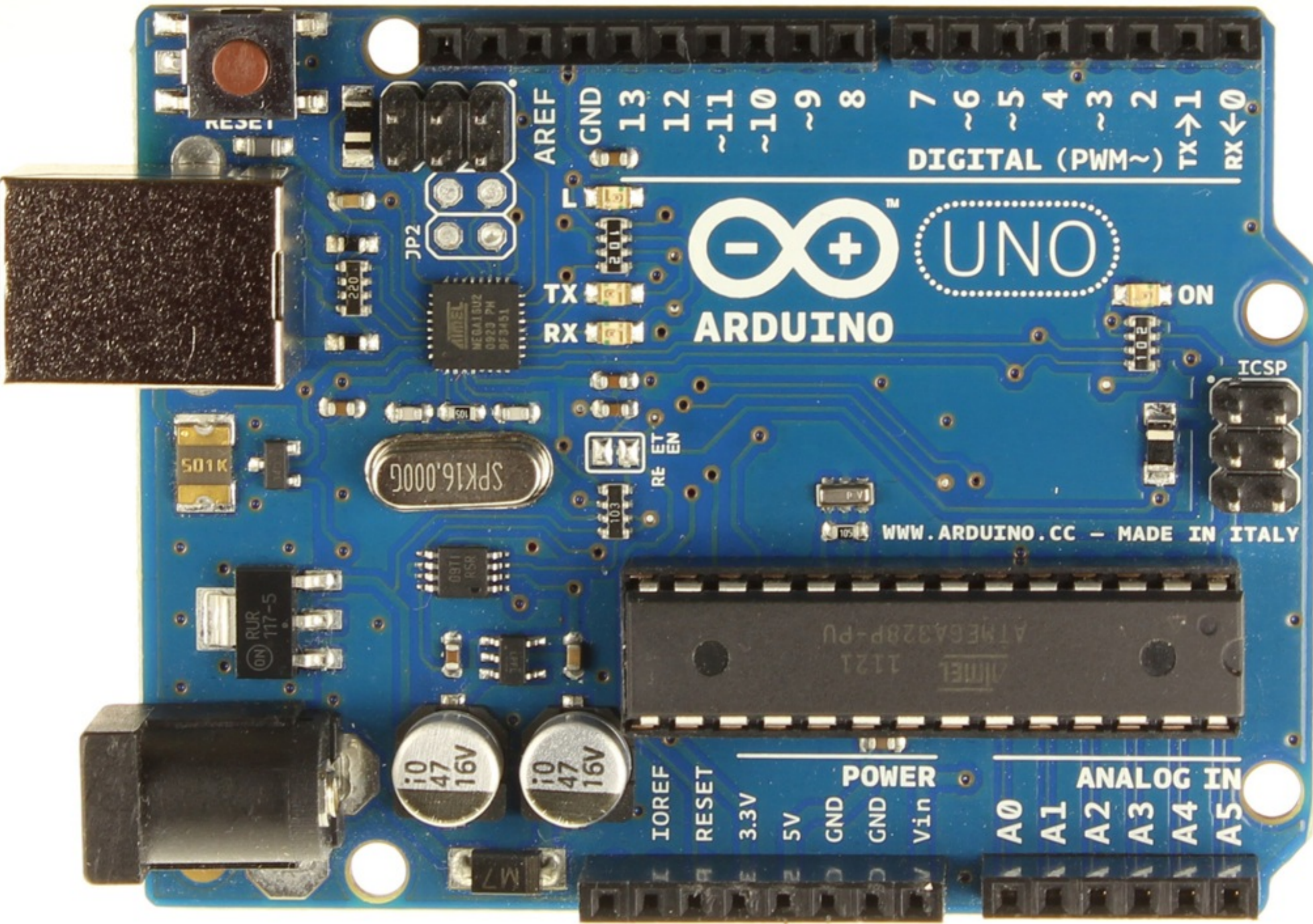
- Entender y utilizar los ALGORITMOS
- Concebir, proponer, implementar y evaluar un pequeño PROYECTO
- Empezar a distinguir entre señales CONTINUAS y DISCRETAS



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/>



RESET

JP2

AREF

GND

13

12

~11

~10

~9

8

7

6

~5

~4

3

2

1

TX

←0

RX

DIGITAL (PWM ~)



UNO

ARDUINO

ON

TX

RX

ICSP

SPK16.000G

RE ET EN

WWW.ARDUINO.CC - MADE IN ITALY

ON RUR 117-5

0971 RSR

ATMEGA328P-PU
1121
ATMEL

i0
47
16V

i0
47
16V

IOREF

RESET

3.3V

5V

GND

GND

Vin

POWER

A0

A1

A2

A3

A4

A5

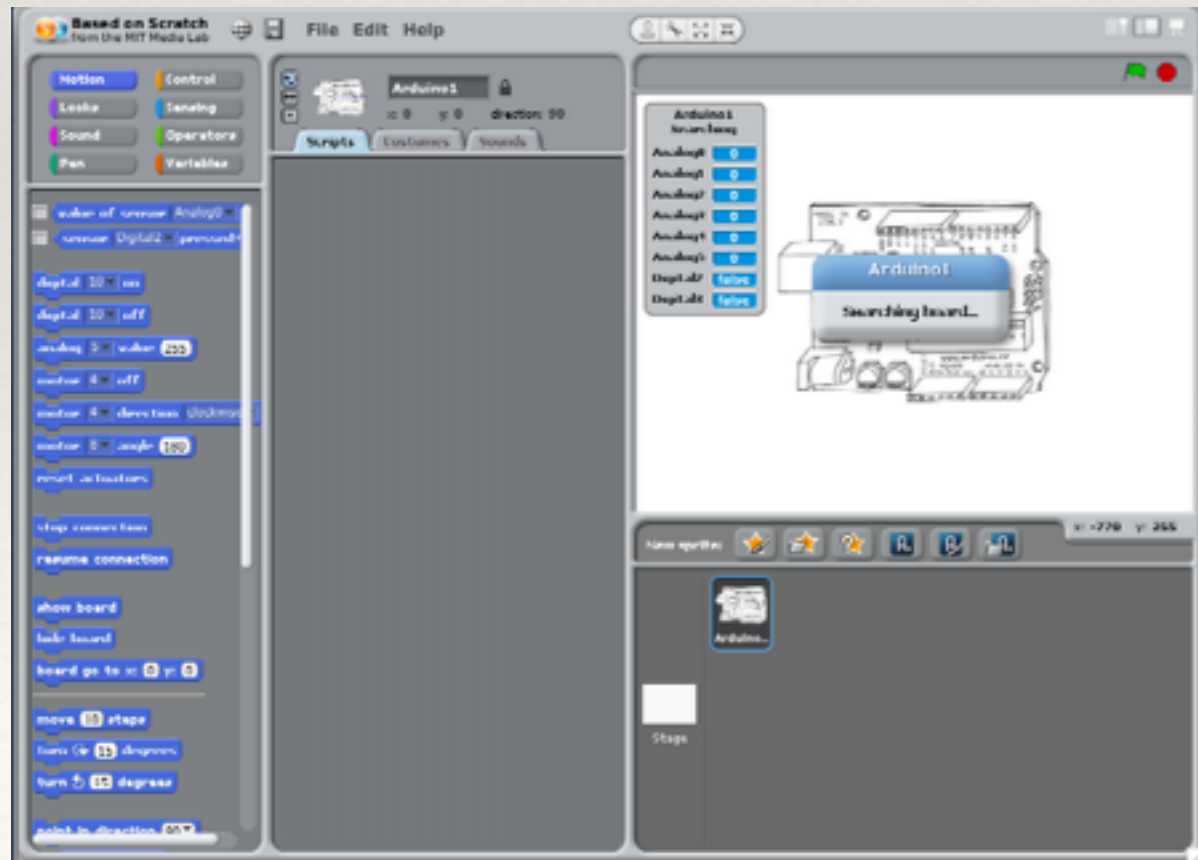
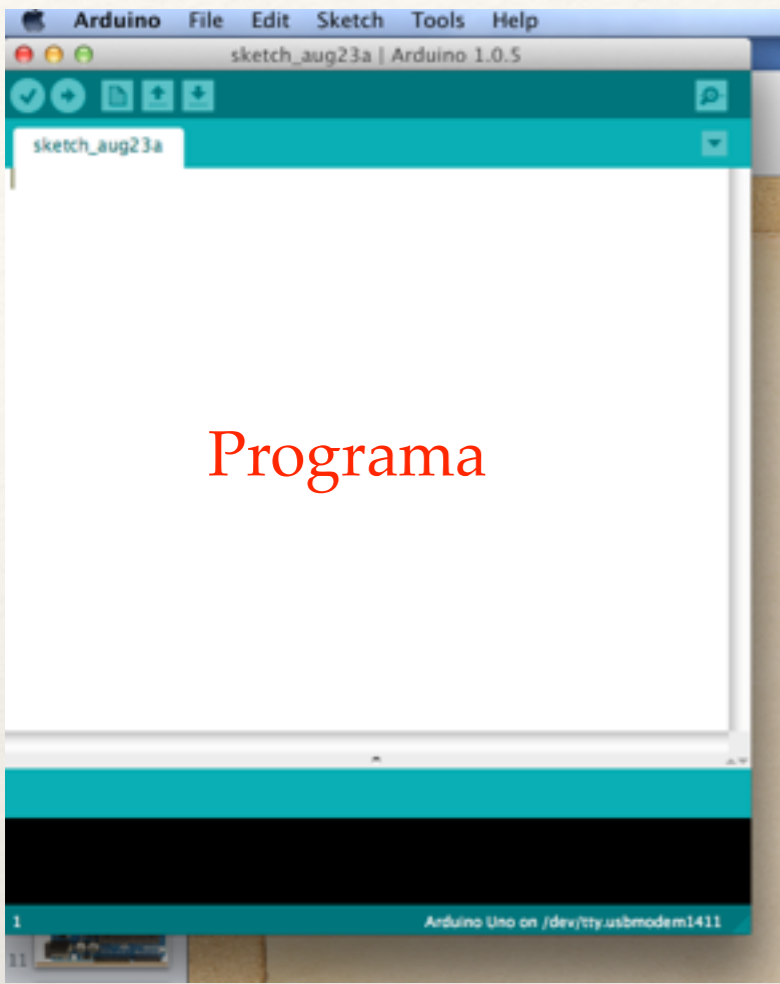
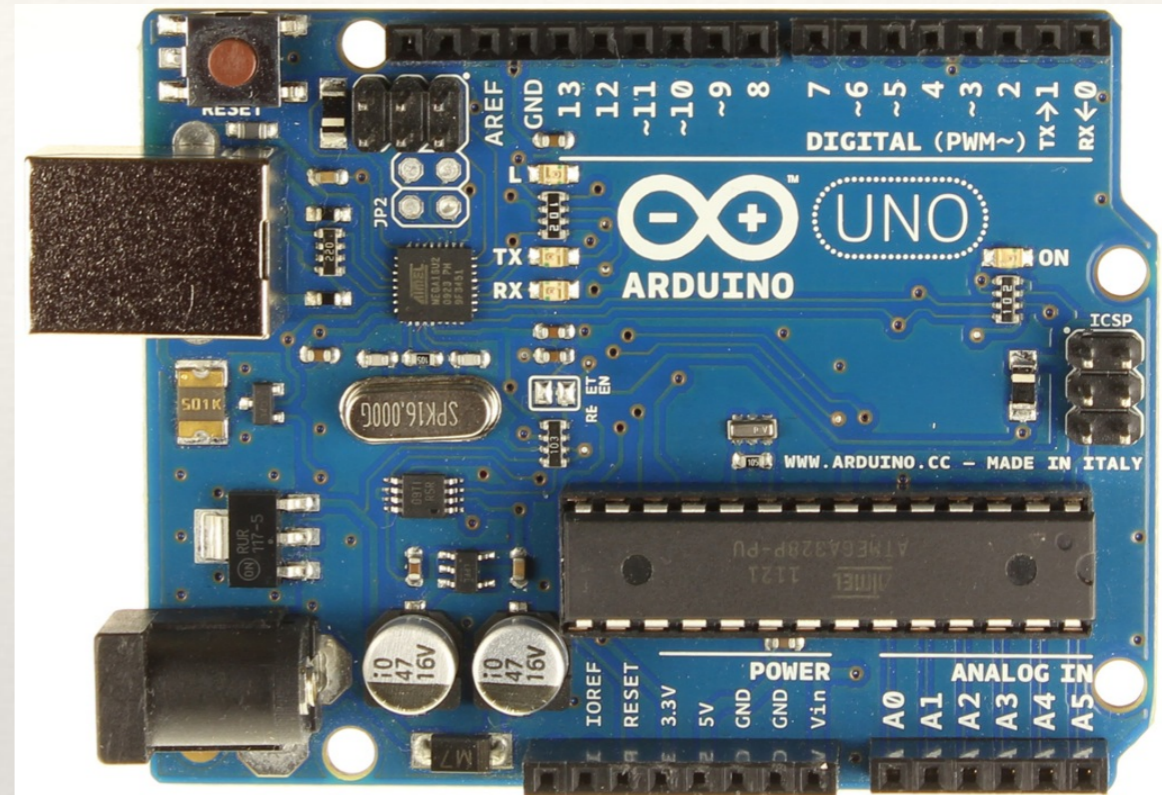
ANALOG IN

Entorno de desarrollo con ARDUINO

IDE de Arduino

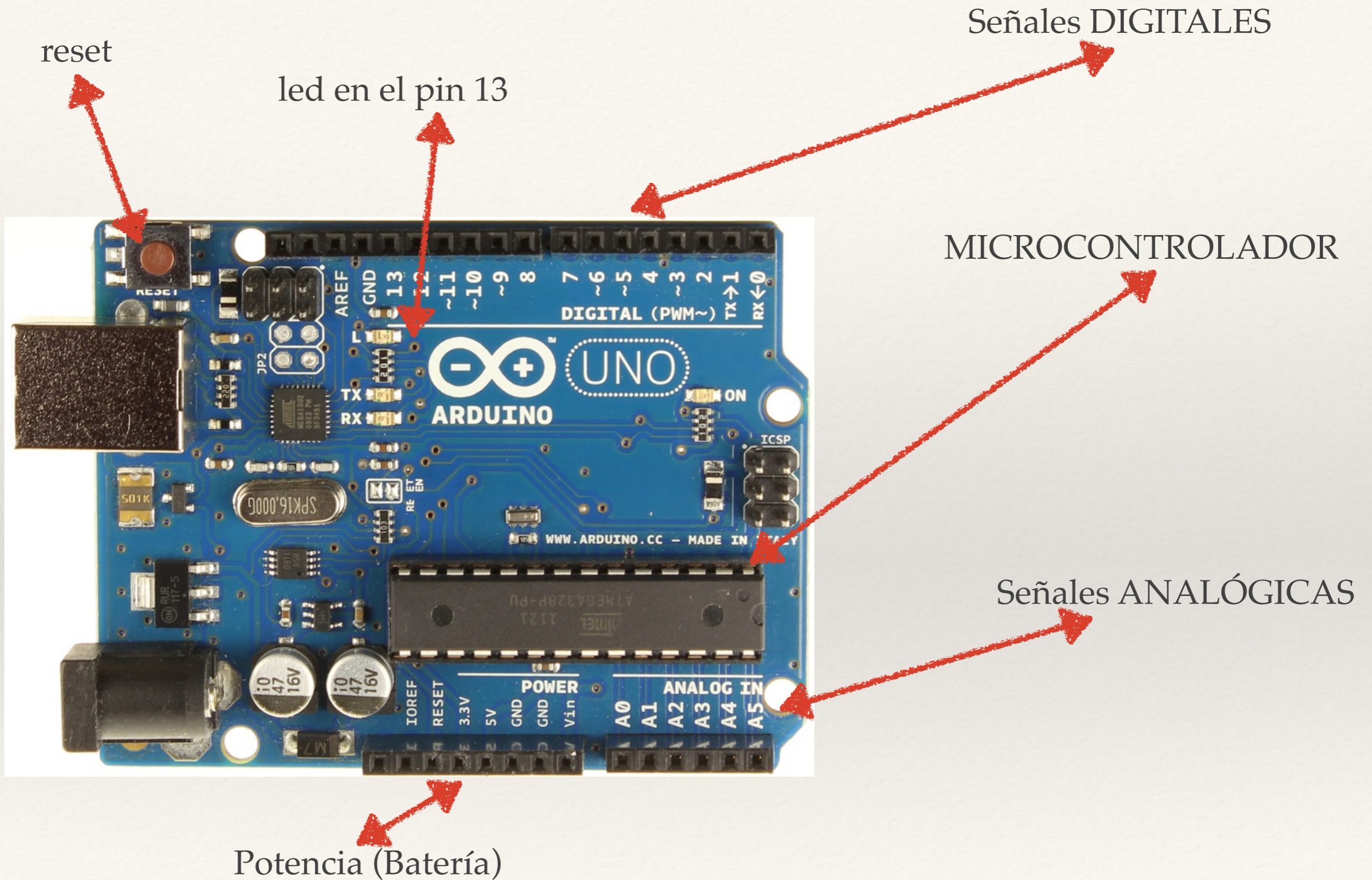
Programa

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

```

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);

}

```

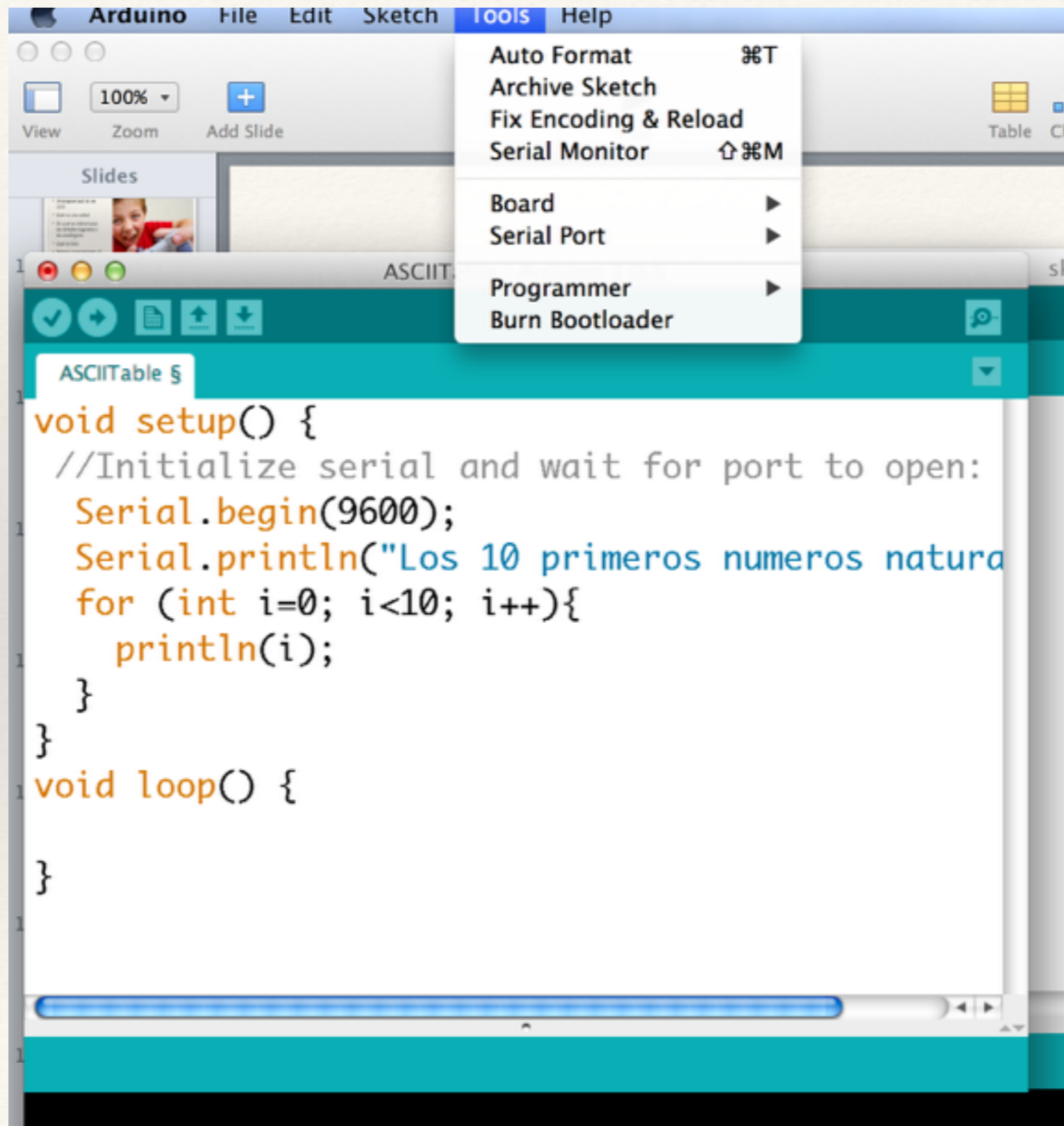
```

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 such as 'Auto Format', 'Archive Sketch', 'Fix Encoding & Reload', 'Serial Monitor', 'Board', 'Serial Port', 'Programmer', and 'Burn Bootloader'. Below the menu, a sketch window titled 'ASCIITable §' contains 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() {  
  
}
```

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%

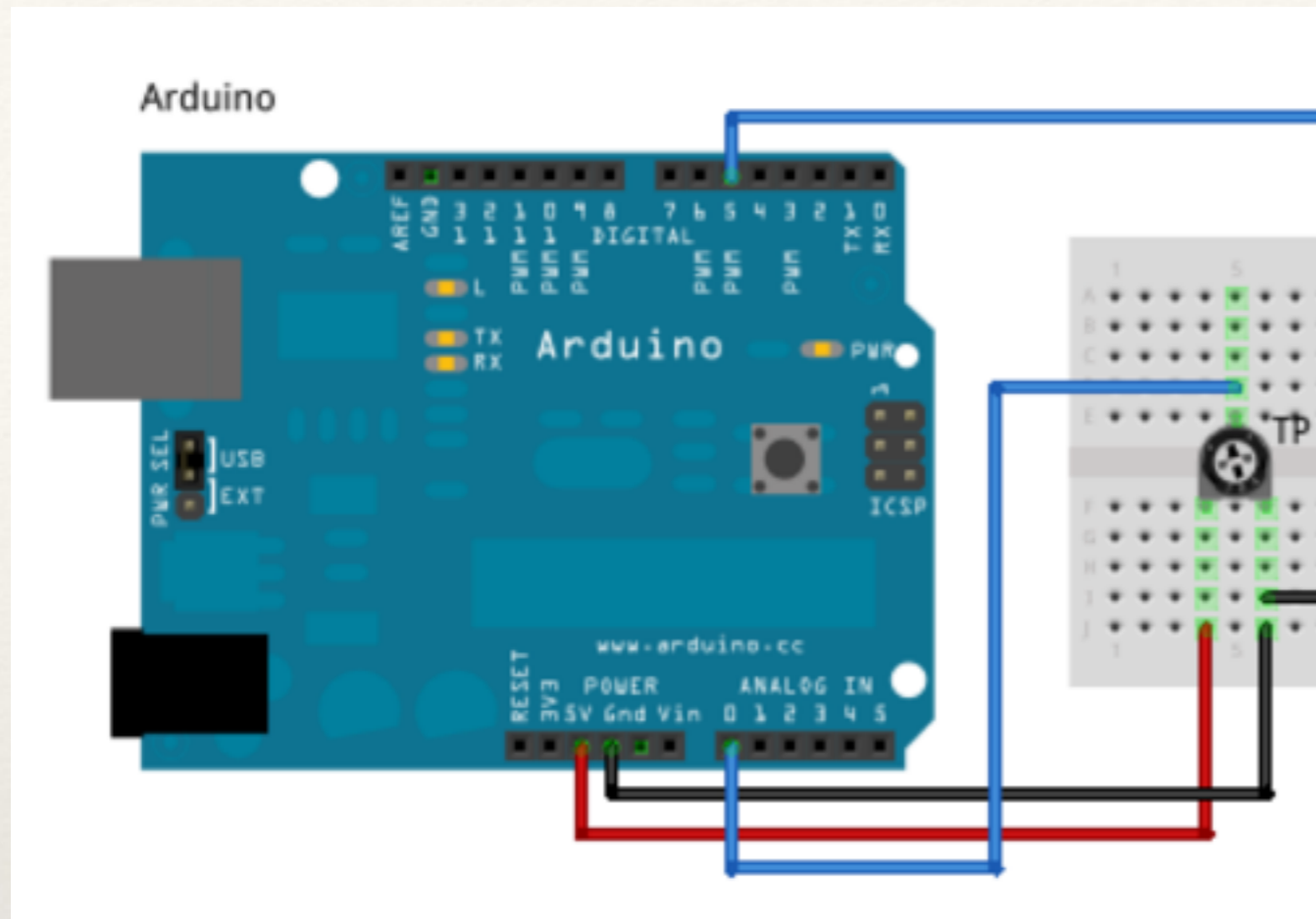


Prog6

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

```
1  int retardo=1000;
2  void setup() {
3    // initialize digital pin 13 as an output.
4    pinMode(13, OUTPUT);
5  }
6
7  // the loop function runs over and over again forever
8  void loop() {
9    for(int i=0;i<4;i++) {
10     digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
11     delay(retardo);        // wait for a second
12     digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
13     delay(retardo);
14     retardo =              ;
15   }
16   //
17   // segunda secuencia
18   //
19   for(int i=0;i<4;i++) {
20     digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
21     delay(retardo);        // wait for a second
22     digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
23     delay(retardo);
24     retardo =              ;
25   }
26 }
```

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() {

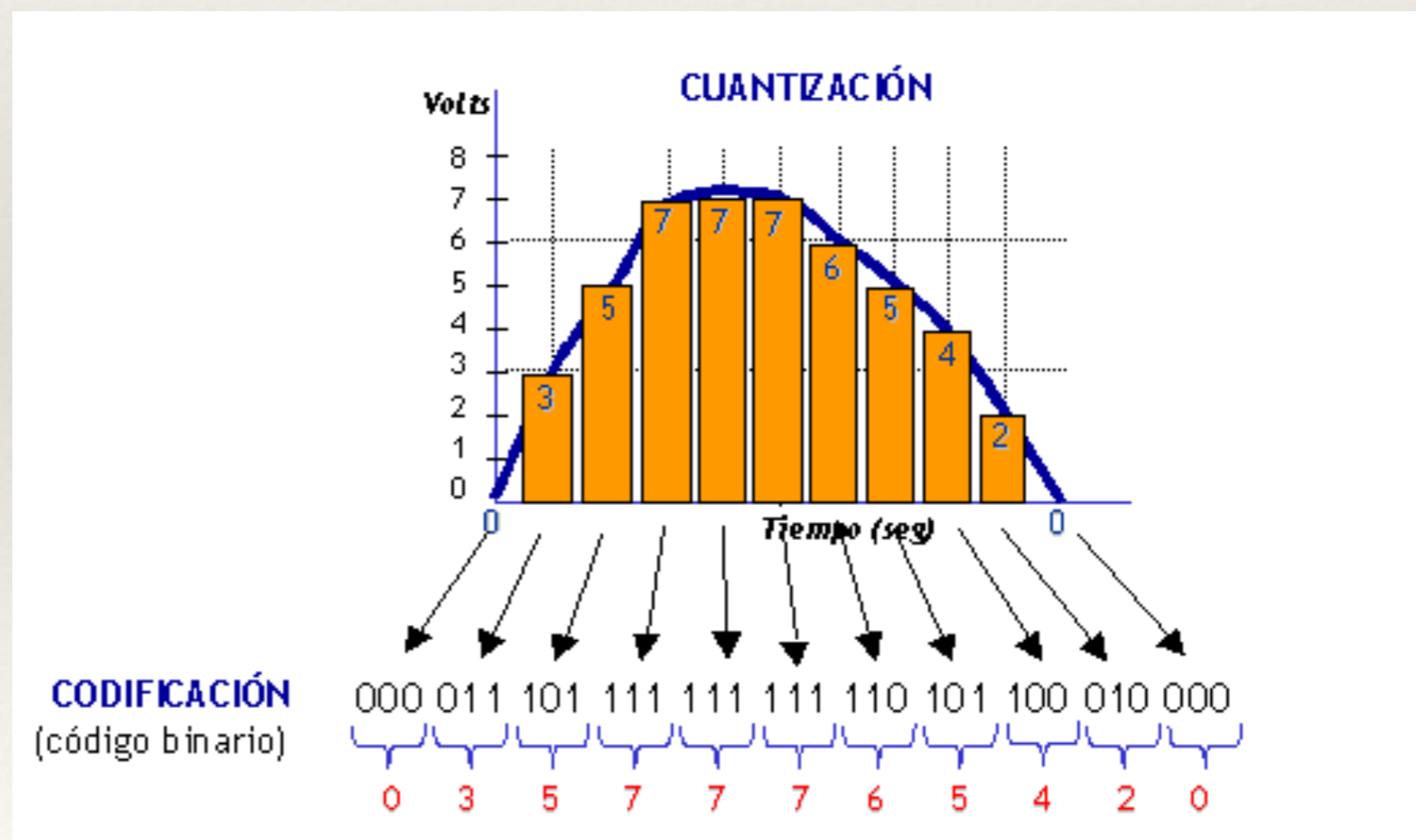
  val = analogRead(analogPin);    // read the input pin
  Serial.println(val);
  delay(400); // debug value
}
```

Las señales analógicas

video



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 |

| | | | | | | | | | | | |
|----------|---|-------|----------|---|-----------|----------|---|---------------|----------|---|-------------------|
| 2^0 | = | 1 | | | | | | | | | |
| 2^1 | = | 2 | 2^{11} | = | 2.048 | 2^{21} | = | 2.097.152 | 2^{31} | = | 2.147.483.648 |
| 2^2 | = | 4 | 2^{12} | = | 4.096 | 2^{22} | = | 4.194.304 | 2^{32} | = | 4.294.967.296 |
| 2^3 | = | 8 | 2^{13} | = | 8.192 | 2^{23} | = | 8.388.608 | 2^{33} | = | 8.589.934.592 |
| 2^4 | = | 16 | 2^{14} | = | 16.384 | 2^{24} | = | 16.777.216 | 2^{34} | = | 17.179.869.184 |
| 2^5 | = | 32 | 2^{15} | = | 32.768 | 2^{25} | = | 33.554.432 | 2^{35} | = | 34.359.738.368 |
| 2^6 | = | 64 | 2^{16} | = | 65.536 | 2^{26} | = | 67.108.864 | 2^{36} | = | 68.719.476.736 |
| 2^7 | = | 128 | 2^{17} | = | 131.072 | 2^{27} | = | 134.217.728 | 2^{37} | = | 137.438.953.472 |
| 2^8 | = | 256 | 2^{18} | = | 262.144 | 2^{28} | = | 268.435.456 | 2^{38} | = | 274.877.906.944 |
| 2^9 | = | 512 | 2^{19} | = | 524.288 | 2^{29} | = | 536.870.912 | 2^{39} | = | 549.755.813.888 |
| 2^{10} | = | 1.024 | 2^{20} | = | 1.048.576 | 2^{30} | = | 1.073.741.824 | 2^{40} | = | 1.099.511.627.776 |

Practique:

Convierta los siguientes números binarios a decimal:

- 10101010
- 10000000001
- 11111111

Convierta los siguientes números decimales a binario

- 467
- 2345
- 1024

Tarea

Prog7

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

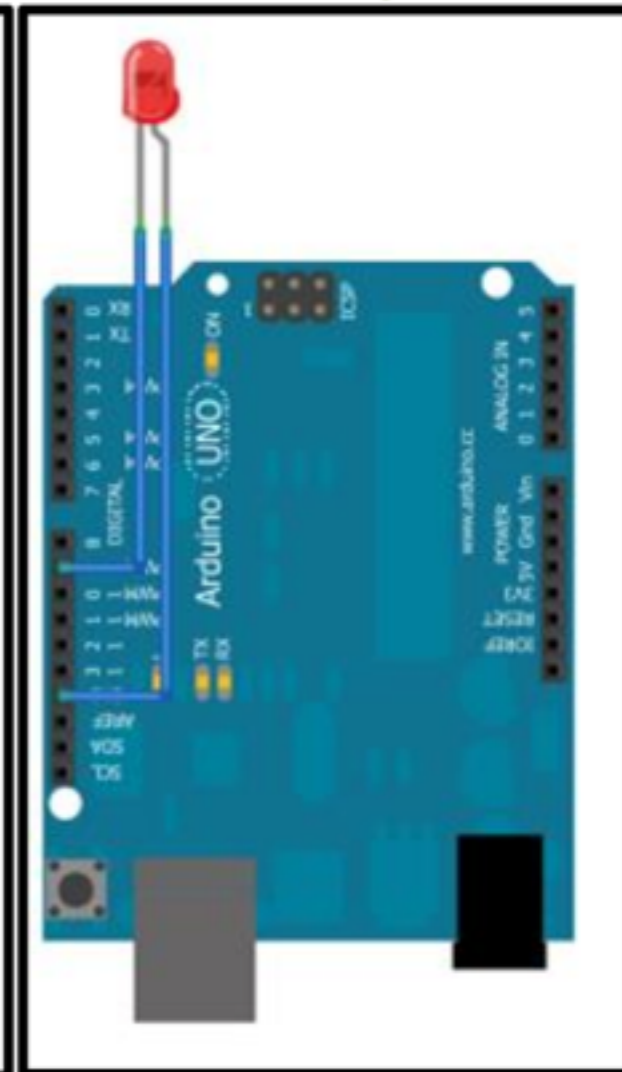
```
/* www.dynamoelectronics.com
Este ejemplo enciende y apaga un led variando su intensidad, este led no puede ser el del pin13 debido
a su tipo, por lo tanto se recomienda usar un led difuso estandar.
La rutina consta de dos for, el primero aumenta los valores de la variable "a" y el segundo los disminuye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



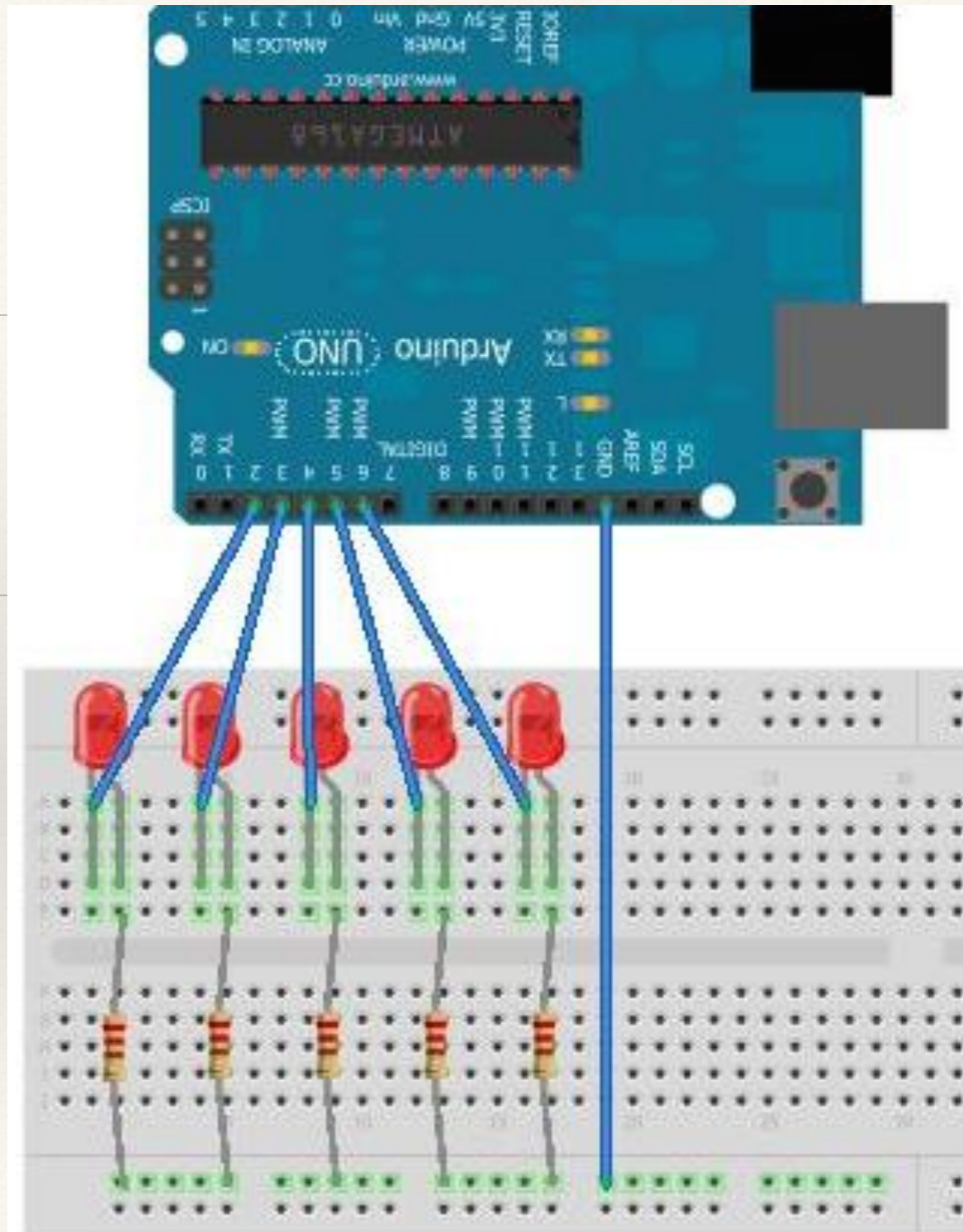
- 2) Investigue qué es y para que sirve PWM

Ejercicio

Prog8

Se requiere que el primer led encienda lentamente hasta brillo máximo, luego apague lentamente. A continuación hace lo mismo el segundo led, y así sucesivamente los 5 leds.

Elabore programa Arduino y esquema en fritzing.



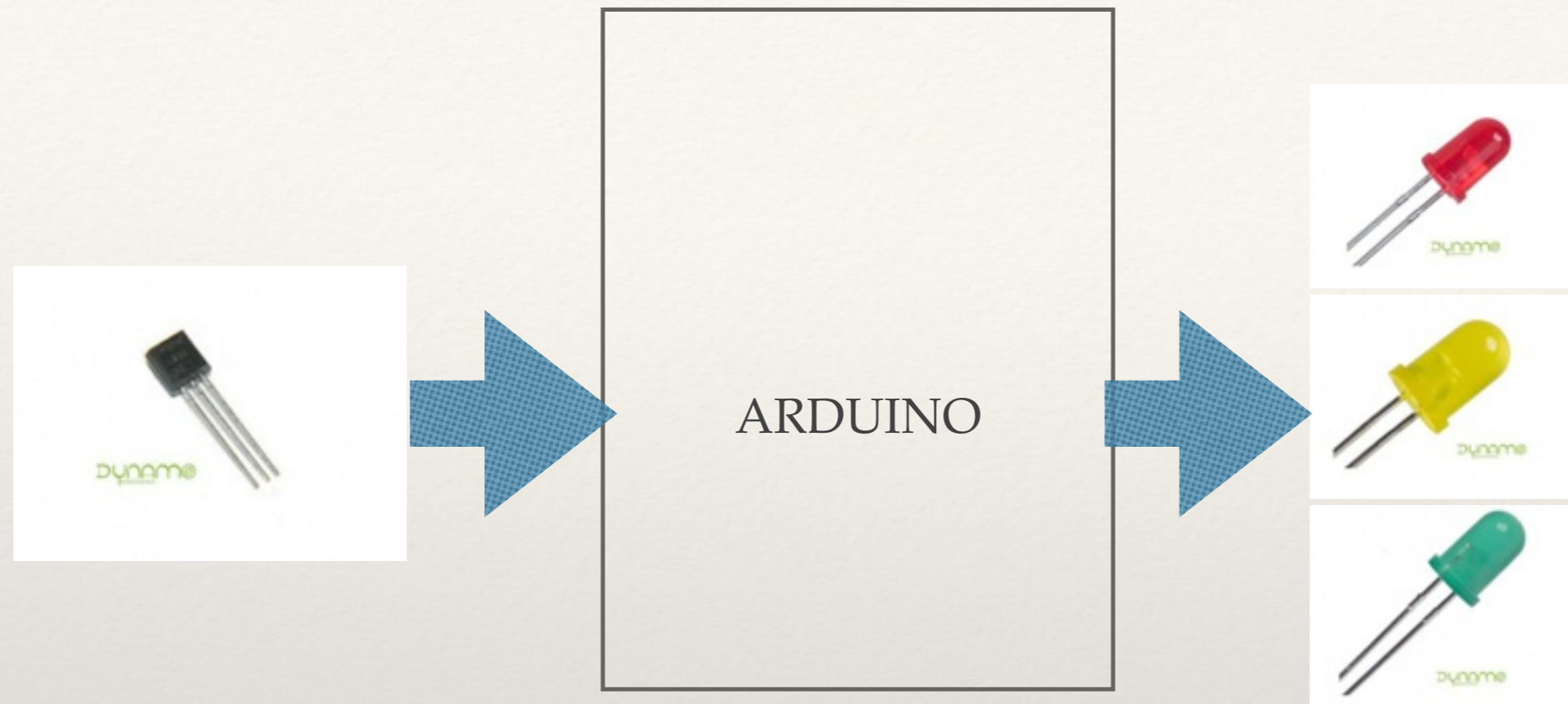
Ejercicio

Prog9

Controlar el brillo de un led por medio del potenciómetro.

Con la misma lectura del potenciómetro, controlar la frecuencia de destello de otro led, con un valor mínimo de 1 destello por segundo y un máximo de 10 destellos por segundo.

Elabore el programa Arduino y el esquema fritzing.



Un sensor de temperatura tiene un factor de conversión de $10 \text{ mV}/^{\circ}\text{C}$. (Simule el sensor con su potenciómetro!!!!). Se requiere que el led verde esté encendido para temperaturas inferiores a 20°C . El led amarillo debe encender cuando se exceda la temperatura de 20°C pero no pase de 25°C . El led rojo debe encender cuando la temperatura exceda de 25°C .
Elaborar programa Arduino y esquema fritzing con PCB