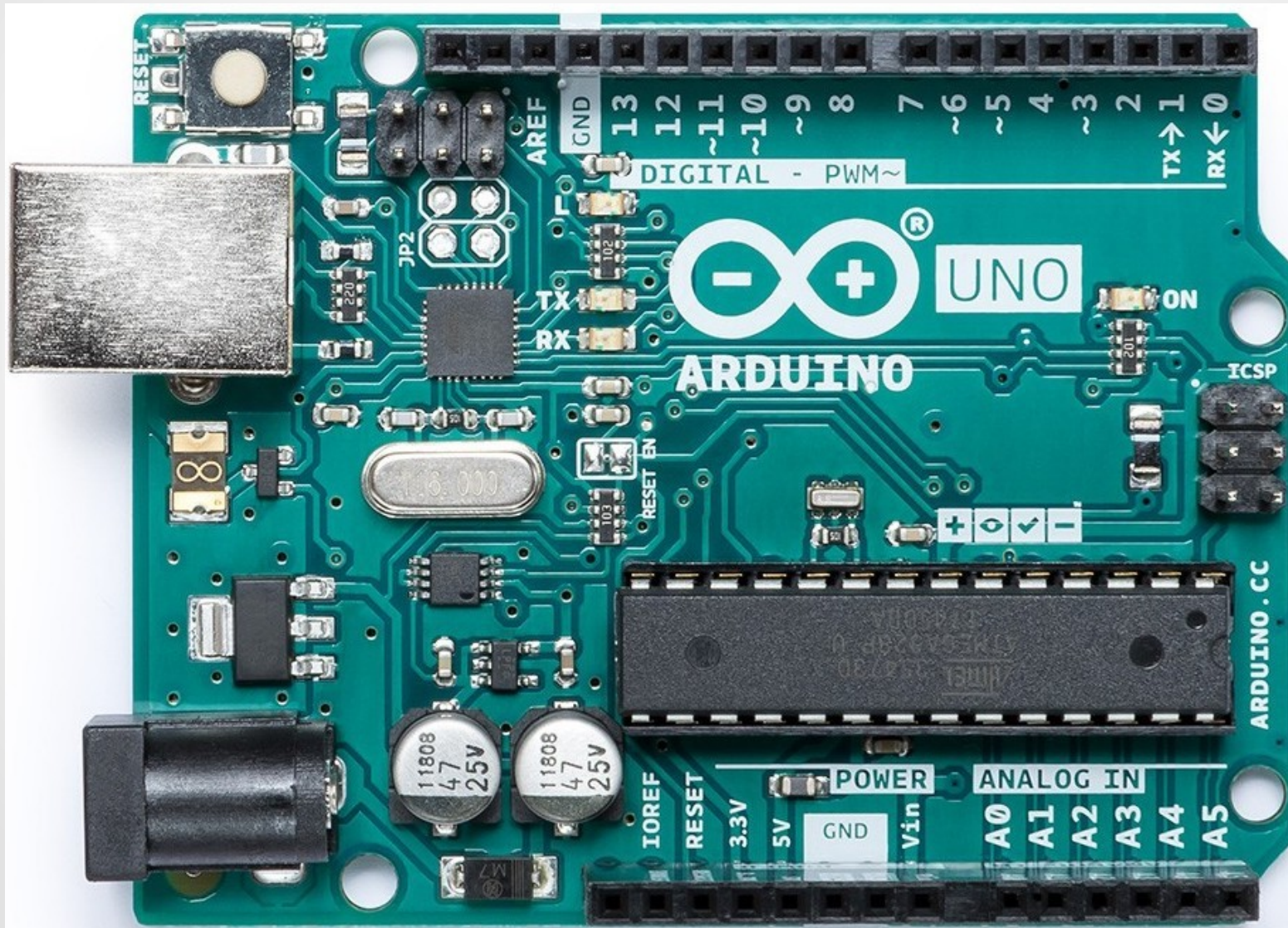


# Introducción – que é Arduino?

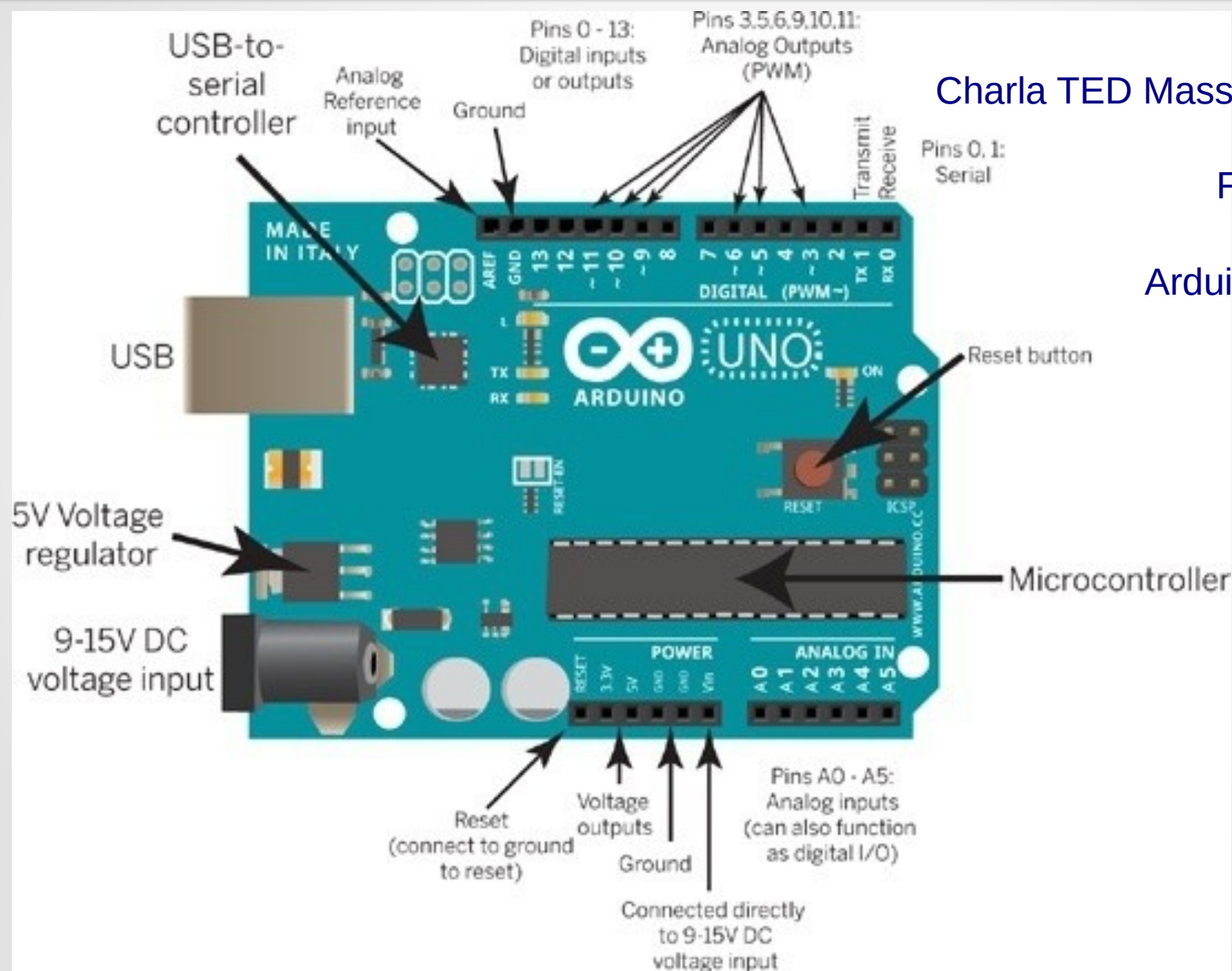


# Introducción – que é Arduino?

Charla TED Massimino Banzi

Familia Arduino

Arduino UNO rev 3



open source  
hardware

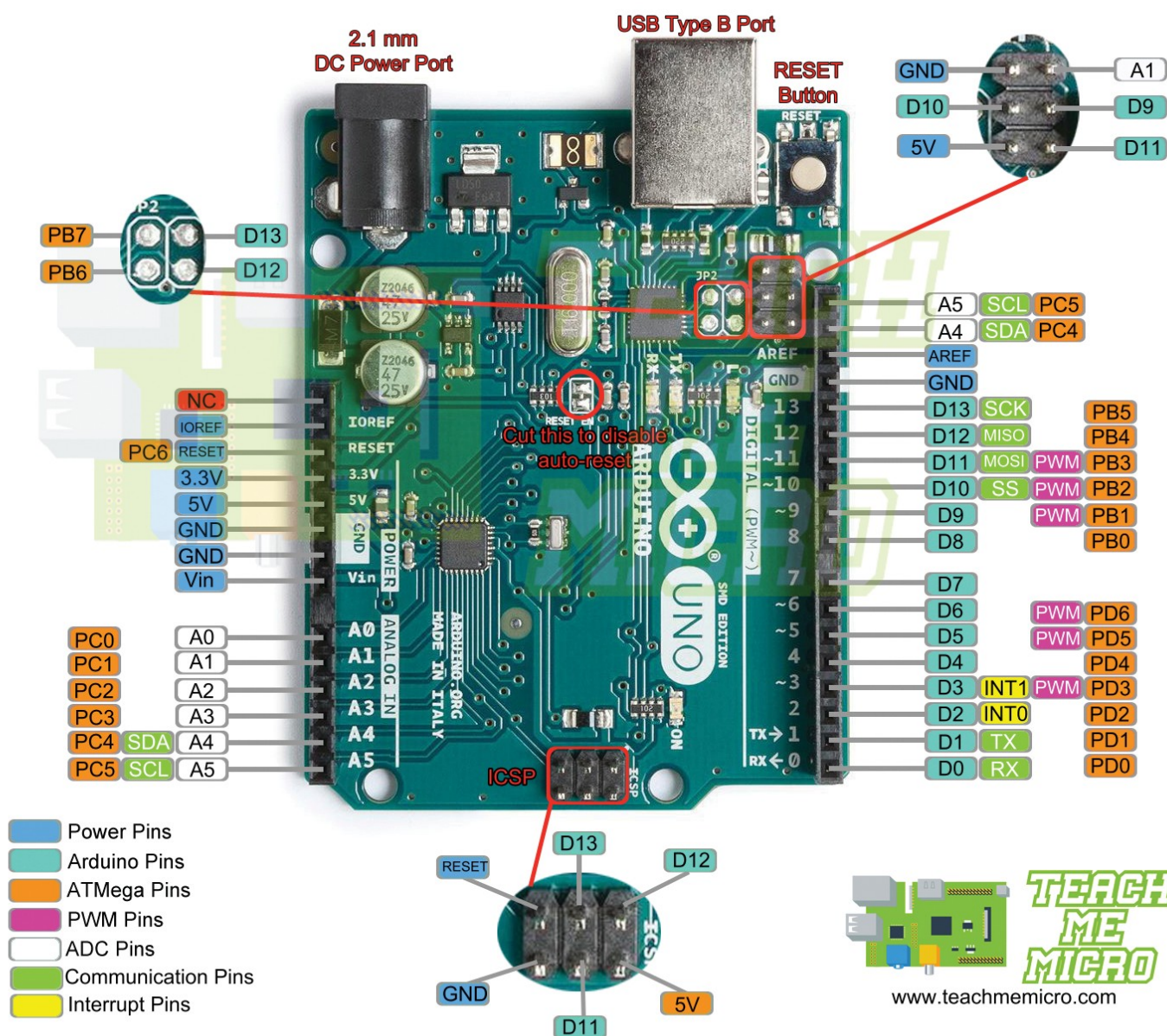


ARDUINO

Victor . Álvarez 201911



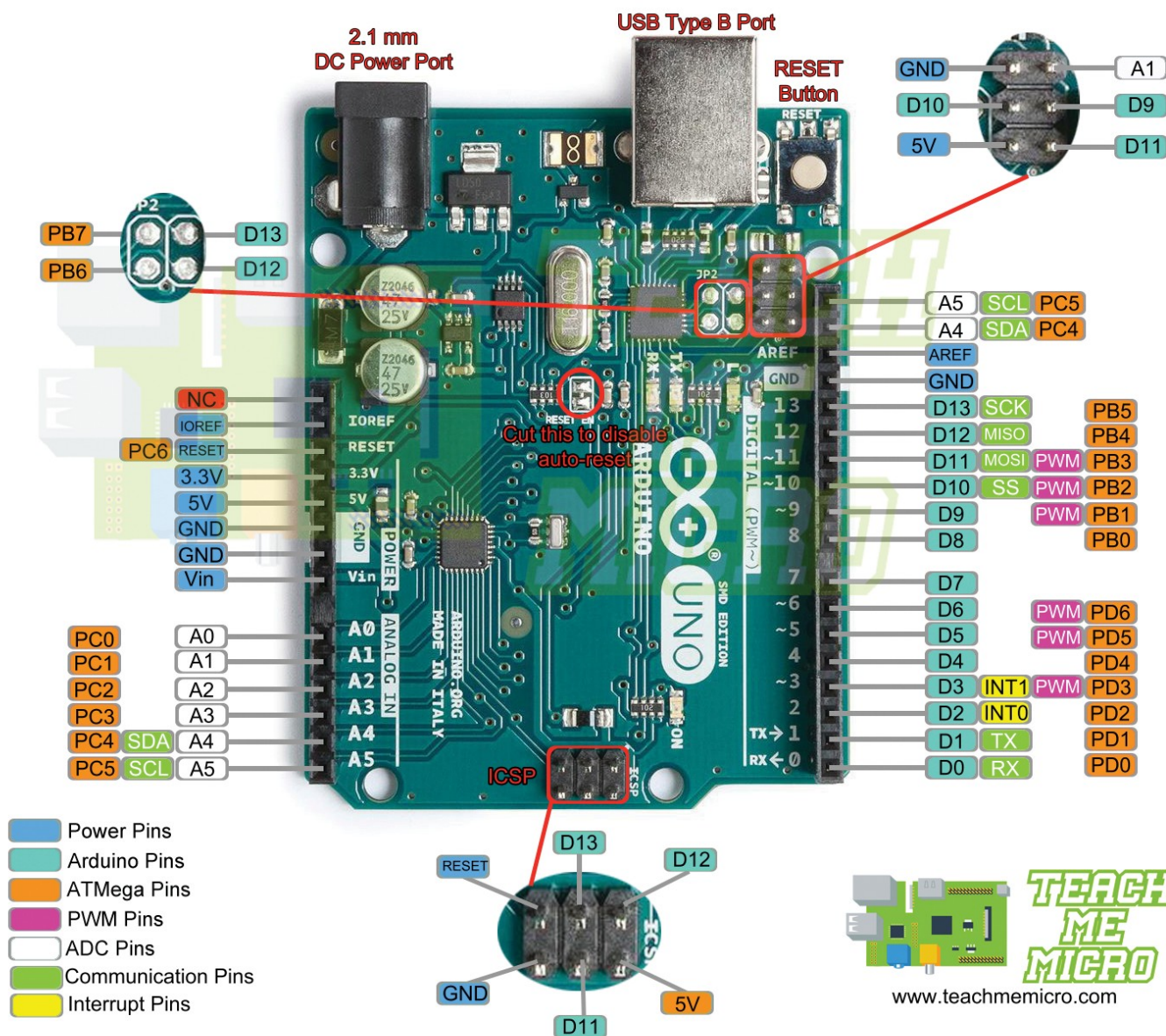
# Introducción – pinout Arduino



## Alimentación:

- USB (5 V)
- jack DC (7 a 12 V)
- Vin (conectado á entrada de alimentación ~1A)
- 3.3 V (máx 50 mA)
- 5 V (máx 300 mA)
- GND (ref 3x)
- AREF (ref alternativa, máx 5 V, 1024 valores)
- Reset

# Introducción – pinout Arduino



## E/S

### Dixitais:

- Pins 0 a 13  
0 a 5V, 20 mA  
Low: 0 a 2V  
High: 3 a 5V

### Entradas Analógicas

- Pins A0 – A5  
0 a 5V, 20 mA prec 1024  
0 a 3V, 50 mA prec 1024

### Saídas PWM

- ~ Pins 3, 5, 6, 9, 10, 11  
0 a 5 V, 20 mA prec 256

### Comunic. Serie TX/RX

- Pins 0 e 1

### ICSP

- 6 pins para comunicarse directamente co proc. Atmega328
- 6 pins para programar o USB





# Introducción – linguaxes de programación

- **Linguaxes de bloques:**

Ideais para iniciarse na programación, evitando a dificultade de editar texto e por tanto os erros sintácticos tan frustrantes ao inicio. Pensadas para nenos mais en moitos casos son moi recomendables para adultos sen experiencia previa en programación. O máis representativo é Scratch, desenvolvido polo MIT.

- Scrathc4Arduino (S4A) <http://s4a.cat>
- Ardublock: <http://blog.ardublock.com/>
- Snap4Arduino (snap): <http://snap4arduino.rocks/>
- Mblock: <https://www.mblock.cc>

- **Linguaxes baseados en texto:**

Son máis flexibles e dispoñen de máis potencia para programar. En troca a curva de aprendizaxe é máis empinada. Dispoñen de moitas librerías preprogramadas que axudan a interaccionar con elementos máis complexos, como shields, sensores, displays, motores, etc. A máis empregada é Sketch (C de Arduino).

- C#, .Net, Python, Node.js, Java, etc
- Sketch (C de Arduino): <https://www.arduino.cc/en/Tutorial/Sketch>

# Introdução – blocos vs código

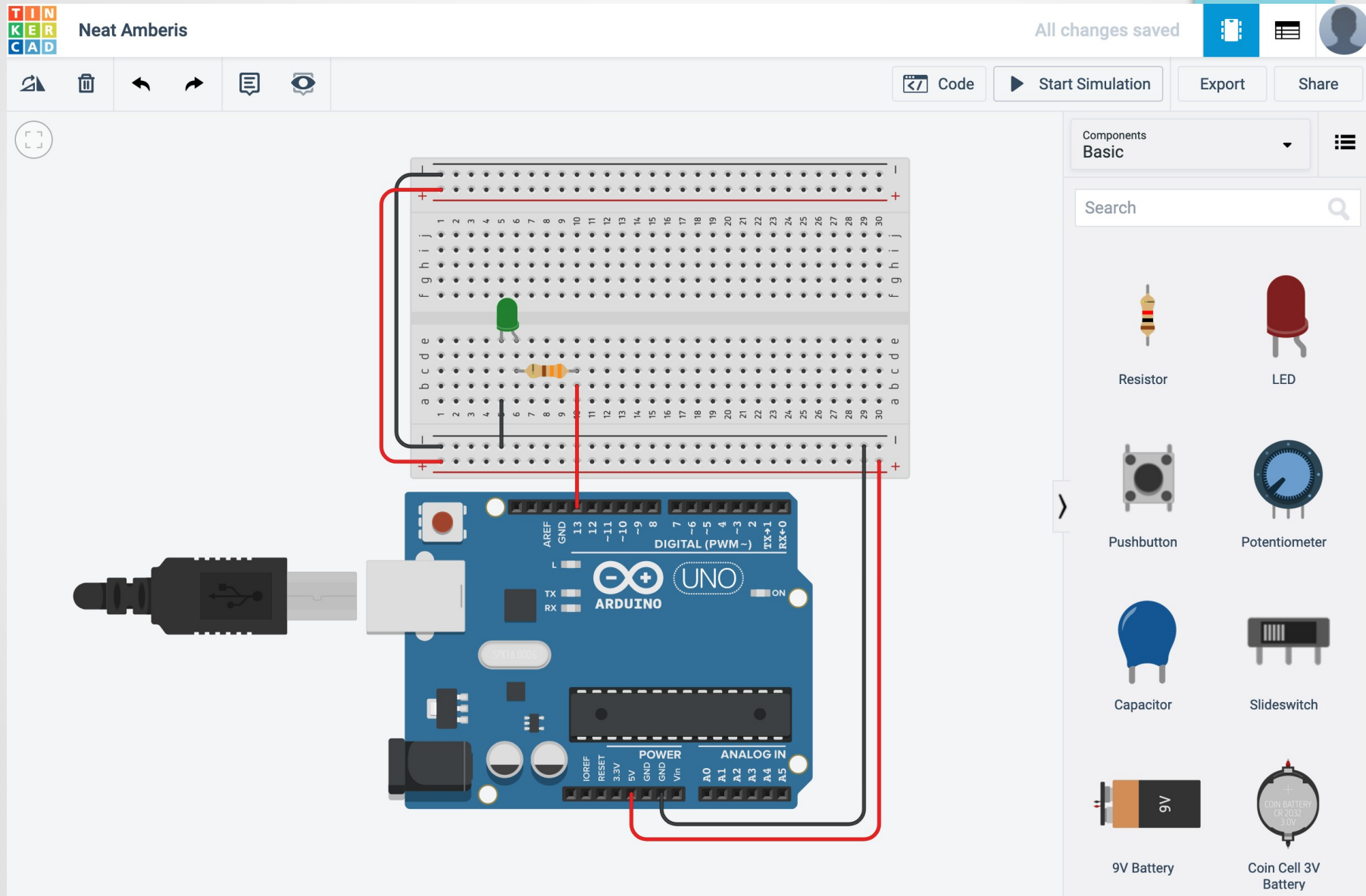
The screenshot displays the mBlock5 IDE interface. The top bar includes the mBlock logo, file management options (Ficheiro, Editar, Sem título, Guardar), and user options (Tutoriais, Sugestão, profile icon). The left sidebar contains a workspace with a panda icon, a 'Carregar' button, and a 'Dispositivos' panel showing an Arduino Uno. The main workspace is divided into two panels: 'Blocos' (Blocks) and 'Código' (Code).

**Blocos Panel:** A sequence of blocks is visible, including 'ler pino digital' (9), 'ler pino analógico (A)' (0), 'ler o pino de pulso' (13) with a 'Timeout' of 2, 'definir a saída do pino digital' (9), 'definir a saída PWM' (5) with 'como' (0), 'tocar pino' (9) with 'nota' (C4), 'definir ângulo do pino do servo' (9), 'suspender pino' (2) with 'Modo' (borda), and 'Não suspenda o pino' (2).

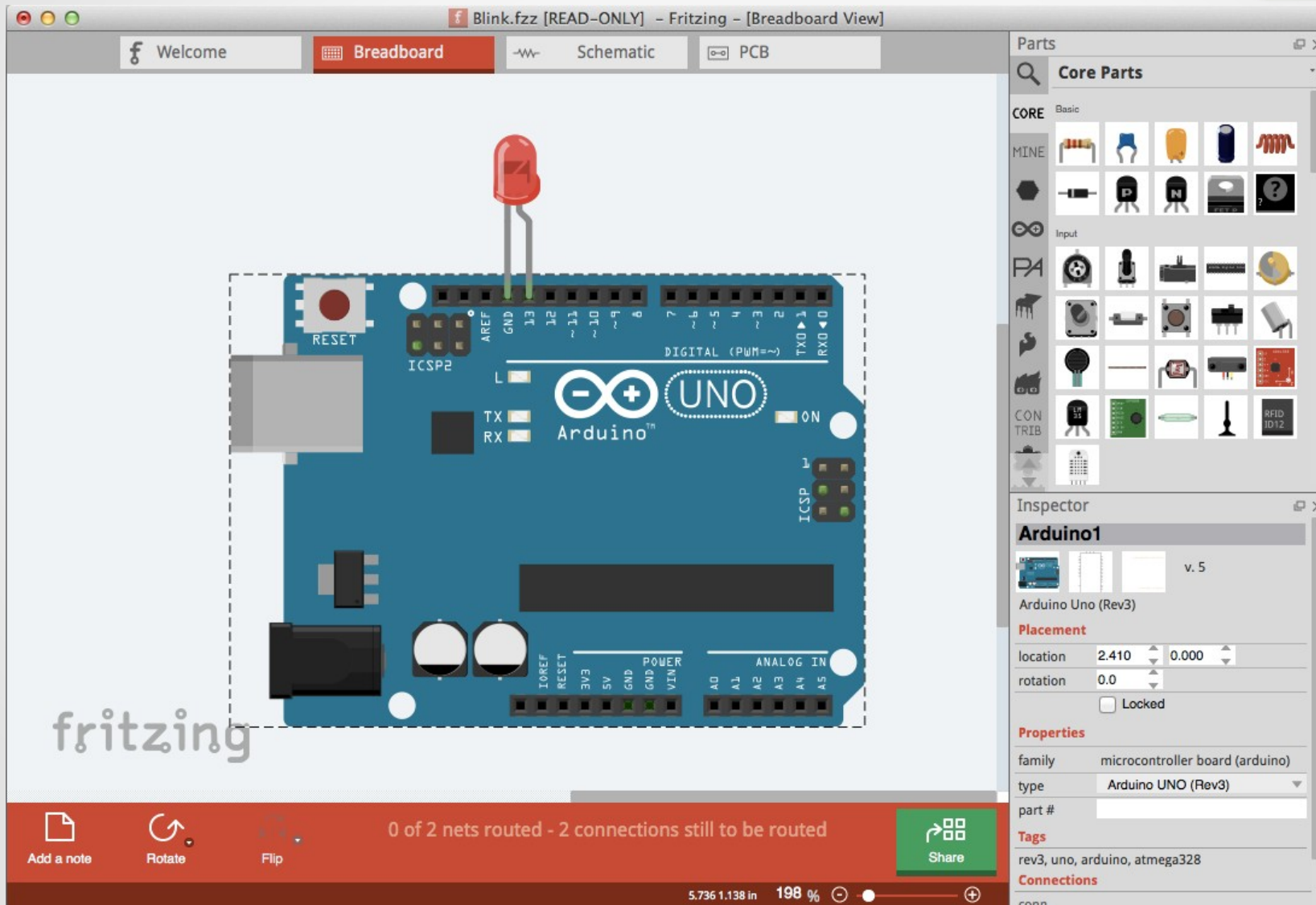
**Código Panel:** The Arduino C code is displayed, generated by mBlock5. It includes headers for `Arduino.h`, `Wire.h`, and `SoftwareSerial.h`. The code defines a `_delay` function, a `setup` function that configures pin 13 as an output and enters a loop with `digitalWrite` and `_delay` calls, and a `loop` function that calls `_loop`.

```
1 // generated by mBlock5 for <your product>
2 // codes make you happy
3
4 #include <Arduino.h>
5 #include <Wire.h>
6 #include <SoftwareSerial.h>
7
8 void _delay(float seconds) {
9     long endTime = millis() + seconds * 1000;
10    while(millis() < endTime) _loop();
11 }
12
13 void setup() {
14     pinMode(13,OUTPUT);
15     while(1) {
16         digitalWrite(13,1);
17         _delay(1);
18         digitalWrite(13,0);
19         _delay(1);
20     }
21     _loop();
22 }
23
24 }
25
26 void _loop() {
27 }
28
29 void loop() {
30     _loop();
31 }
```

# Introducción – emulación de Arduino



# Introducción – emulación de Arduino



Fritzing

Descarga



# Introducción – estructura script Arduino



```
/*
  Blink

  Turns 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 MKR1000 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

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
  */

// 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(500);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

// 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(500);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

## Comentarios:

- Unha liña: // Bla, bla
- Varias liñas: /\* bla, bla \*/

## Delimitadores:

- Sentencia: ;
- Bloque: { bla; bla; }

## Funcións predefinidas:

- void setup() {}
- void loop() {}

## Cabeceira

- sentencias do preprocesador
- chamada a librerías
- parámetros
- funcións do usuario

# Introducción – instalación IDE Arduino

- Sitio de referencia:

[www.arduino.cc](http://www.arduino.cc)

- Na etiqueta 'Software > Downloads' encontramos os instaladores para distintos SSOO.
- O instalador proporciona o IDE para desenvolver en Sketch, xunto coas librerías máis habituais, configura os portos de comunicación (USB) e os exemplos e axudas.
- Referencias para a linguaxe Sketch:
  - <https://www.arduino.cc/reference/pt/>
  - [https://es.wikibooks.org/wiki/Lenguaje\\_de\\_programaci%C3%B3n\\_Arduino](https://es.wikibooks.org/wiki/Lenguaje_de_programaci%C3%B3n_Arduino)