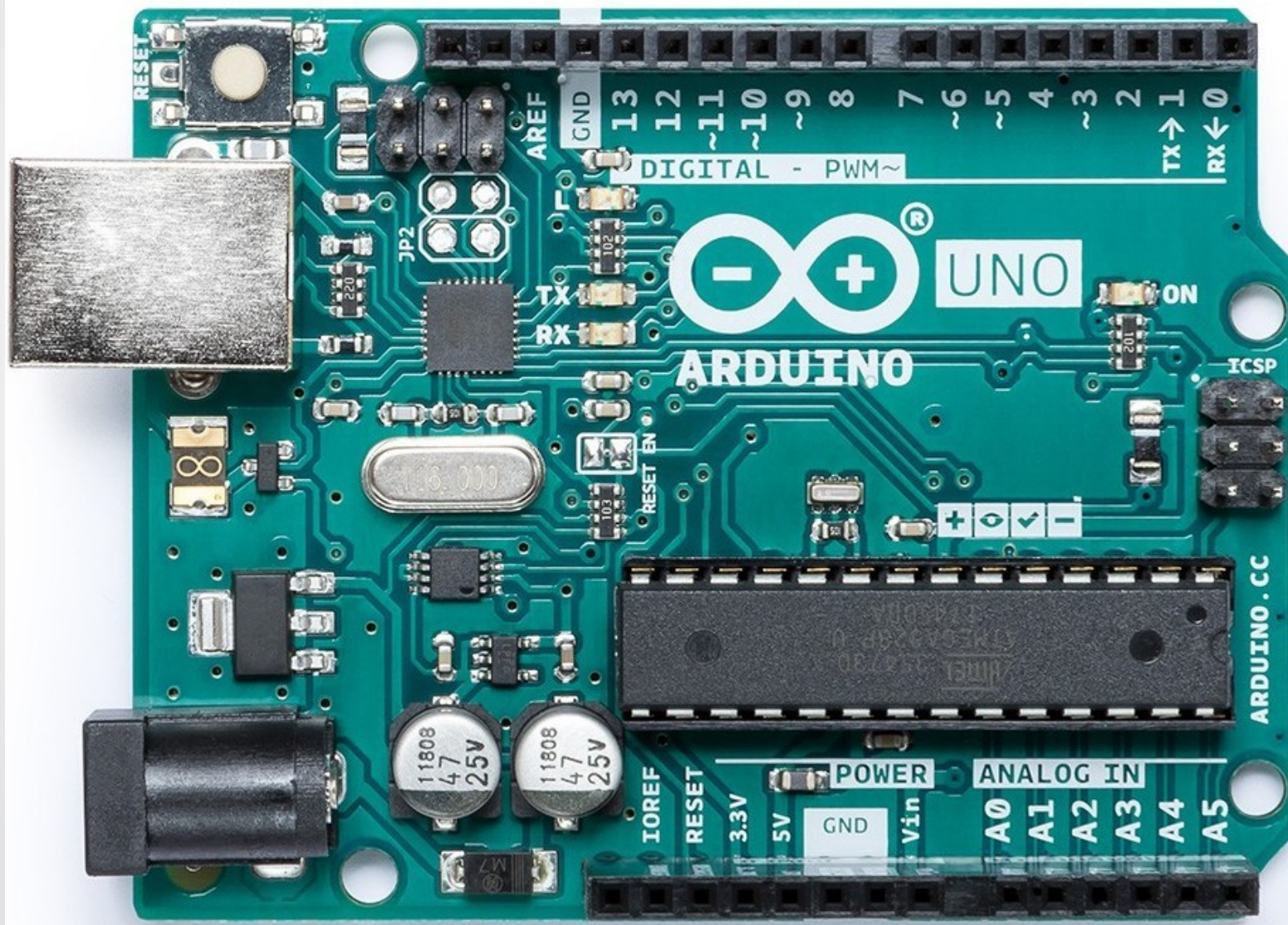
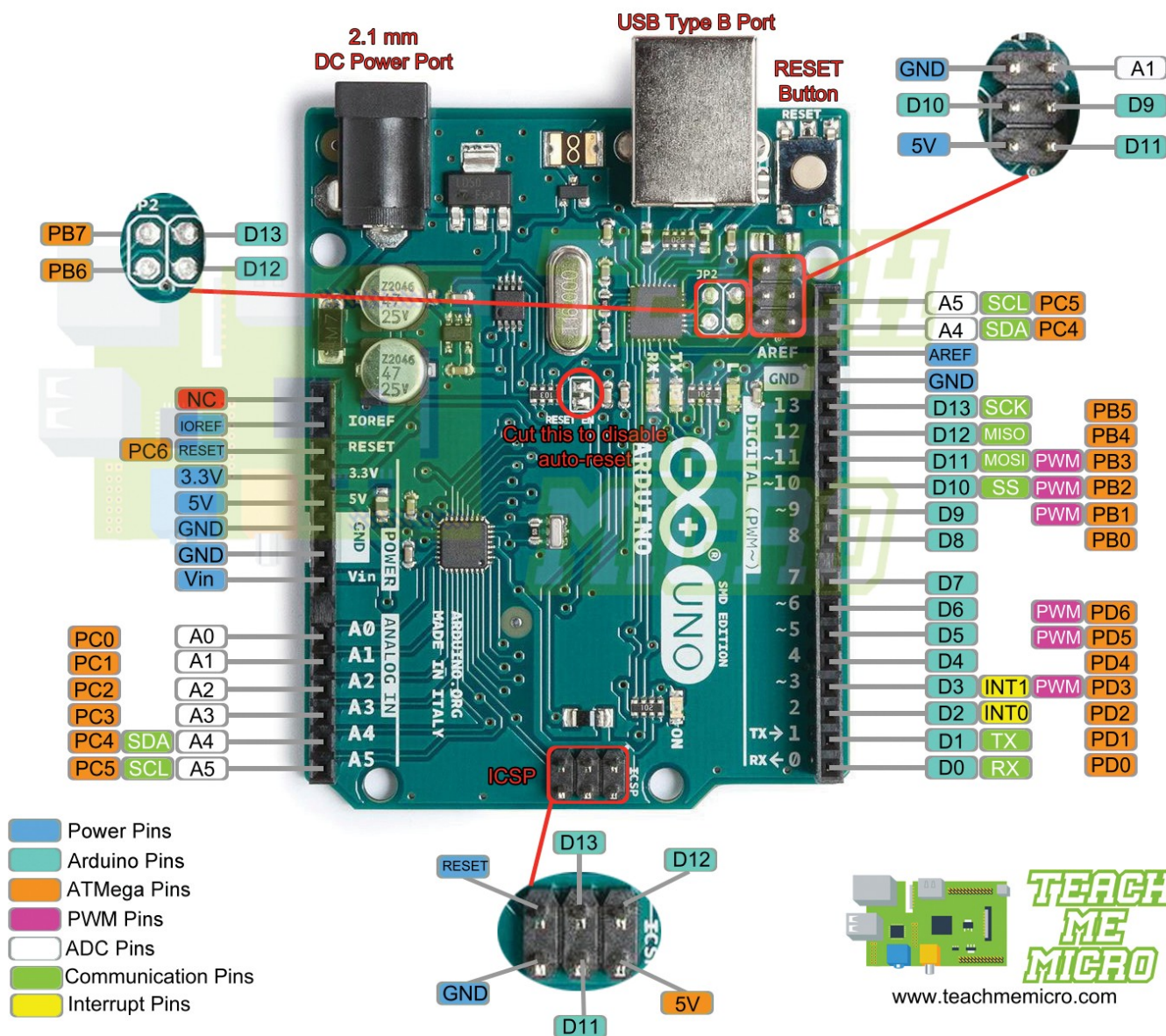


Saídas digitais



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

Saídas dixitais – Encender un LED

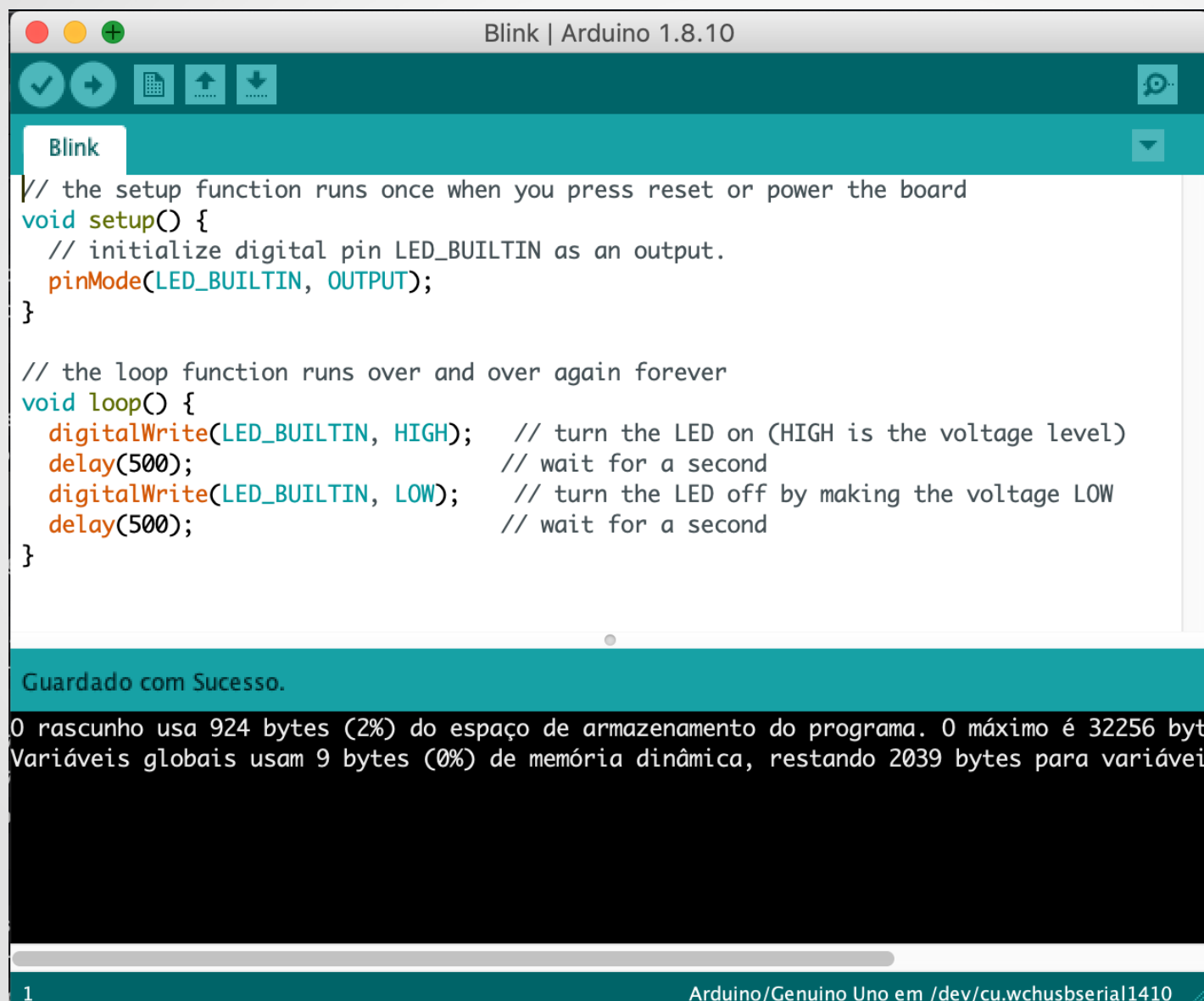
- Para probar as saídas dixitais imos probar cun LED integrado na placa Arduino (pin13) e posteriormente con LEDs na breadboard.
- Empezamos co script 'Blink', dispoñible a través dos menús do IDE en 'Ficheiro > Exemplos > 01.Basics > Blink'.
- Unha vez aberto o script, temos que cargalo na placa, no noso caso Arduino UNO.
- Antes temos que indicarlle ao IDE cal é a placa (en 'Ferramentas > Placa' seleccionamos a nosa placa) e en que porto se pode comunicar con ela (en 'Ferramentas > Porto' seleccionamos o porto).
- Para cargar o script é necesario compilalo antes e posteriormente subilo, os iconos seguintes facilitan isto:



A frecha á dereita realiza a consecutivamente a compilación e carga



Saídas digitais – Encender un LED



```
// 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
}
```

Guardado com Sucesso.

0 rascunho usa 924 bytes (2%) do espaço de armazenamento do programa. O máximo é 32256 bytes. Variáveis globais usam 9 bytes (0%) de memória dinâmica, restando 2039 bytes para variáveis locais.

1 Arduino/Genuino Uno em /dev/cu.wchusbserial1410

Funcións:

pinMode()
digitalWrite()
delay()

Parámetros:

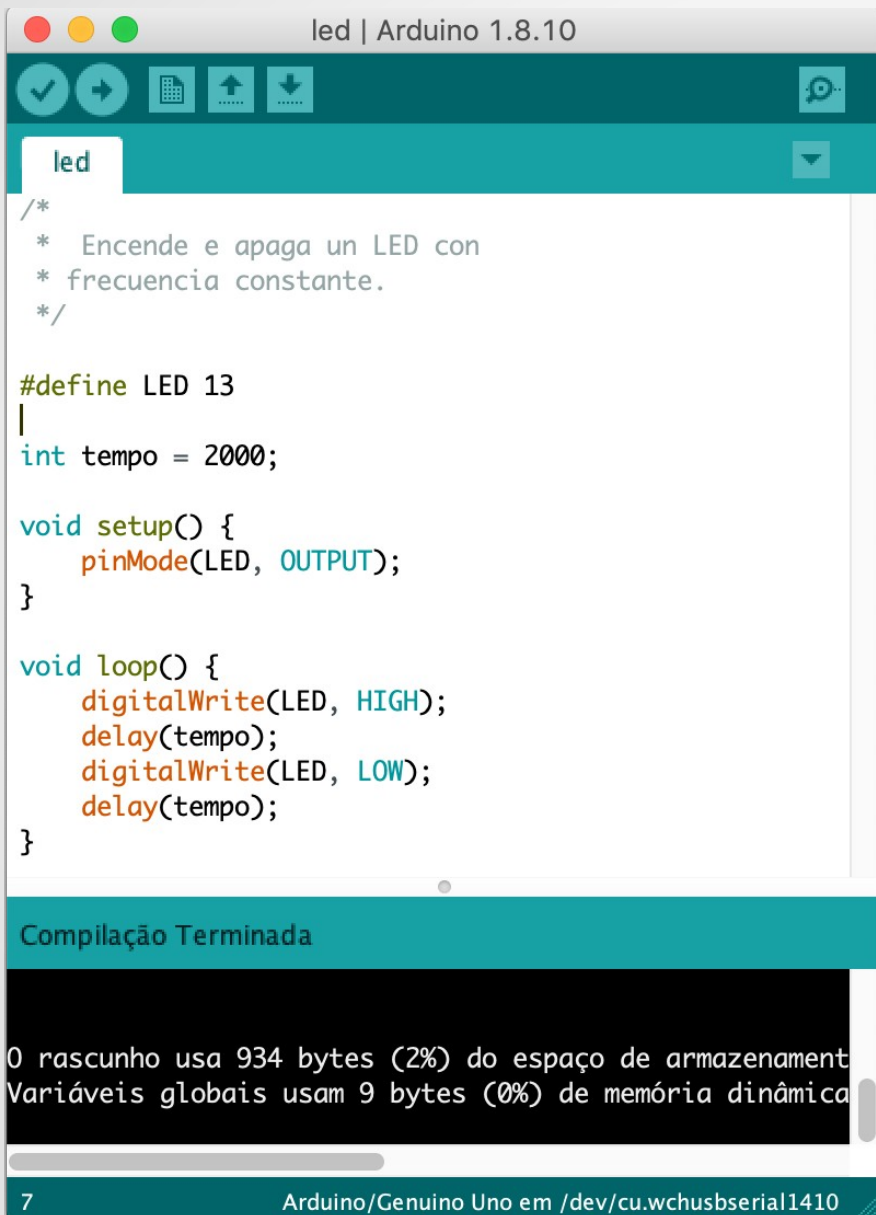
LED_BUILTIN
OUTPUT
HIGH
LOW

Exercicios:

- Cambia o pin a 13
- Cambia a frec. de encendido/apagado do LED



Saídas digitais – Encender un LED



The screenshot shows the Arduino IDE interface with the file named 'led'. The code is as follows:

```
/*
 * Encende e apaga un LED con
 * frecuencia constante.
 */

#define LED 13
|
int tempo = 2000;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(tempo);
  digitalWrite(LED, LOW);
  delay(tempo);
}
```

Below the code editor, a teal status bar indicates 'Compilação Terminada'. The output window shows the following memory usage information:

```
0 rascunho usa 934 bytes (2%) do espaço de armazenament
Variáveis globais usam 9 bytes (0%) de memória dinâmica
```

The status bar at the bottom shows the page number '7' and the serial port path 'Arduino/Genuino Uno em /dev/cu.wchusbserial1410'.



The screenshot shows the Arduino IDE interface with the file named 'led'. The code is as follows:

```
/*
 * Encende e apaga un LED con
 * frecuencia constante.
 */

#define LED 8
int tempo = 10000;

void setup() {
  pinMode(LED, OUTPUT);
}

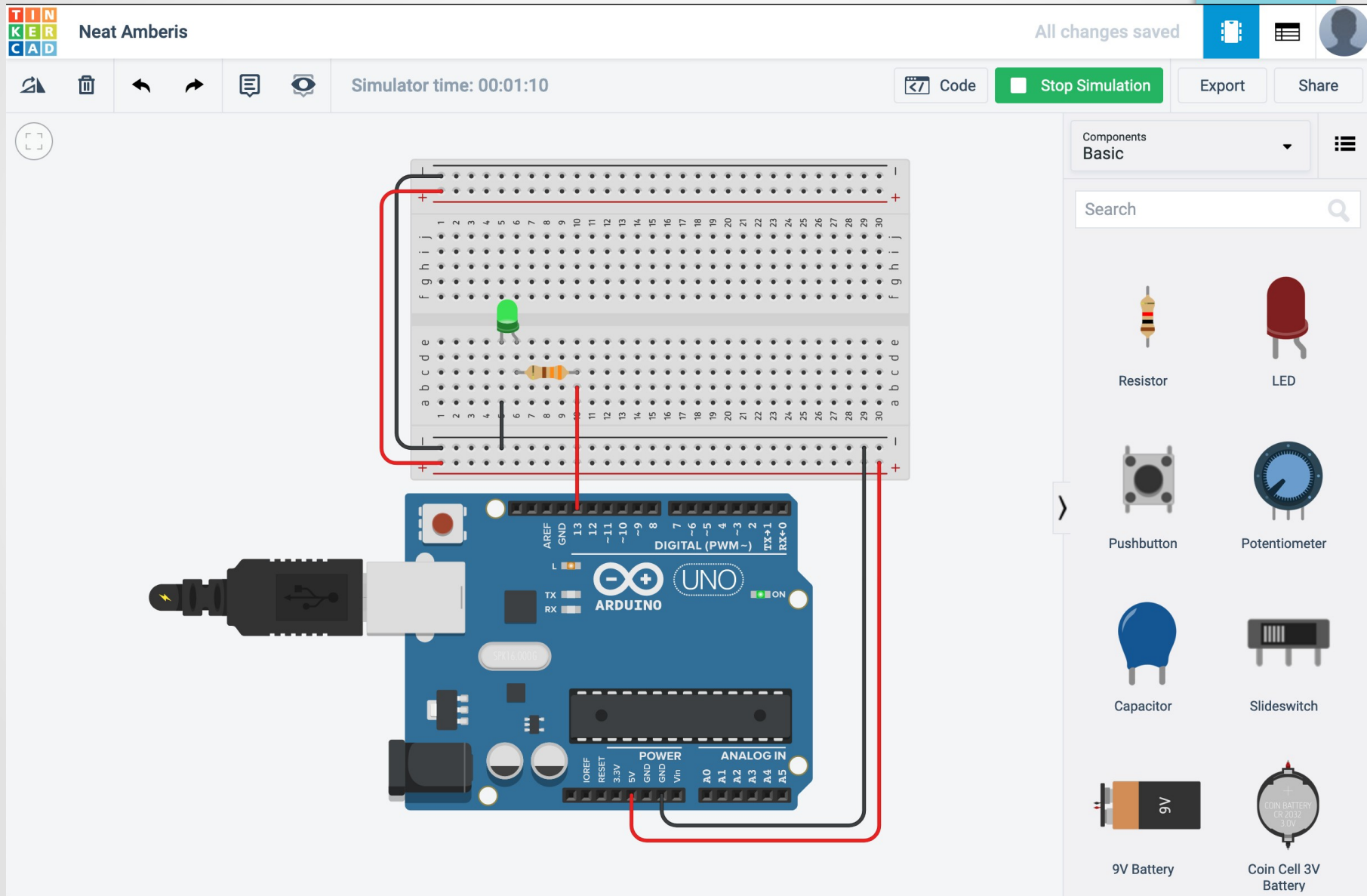
void loop() {
  digitalWrite(LED, HIGH);
  delay(tempo);
  digitalWrite(LED, LOW);
  delay(tempo);
}
```

Below the code editor, a teal status bar indicates 'Guardado com Sucesso.'. The output window shows the following memory usage information:

```
0 rascunho usa 934 bytes (2%) do espaço de armazenament
Variáveis globais usam 9 bytes (0%) de memória dinâmica
```

The status bar at the bottom shows the page number '8' and the serial port path 'Arduino/Genuino Uno em /dev/cu.wchusbserial1410'.

Saídas digitais – Encender un LED



Saídas digitais – Encender un LED

TIN
KER
CAD

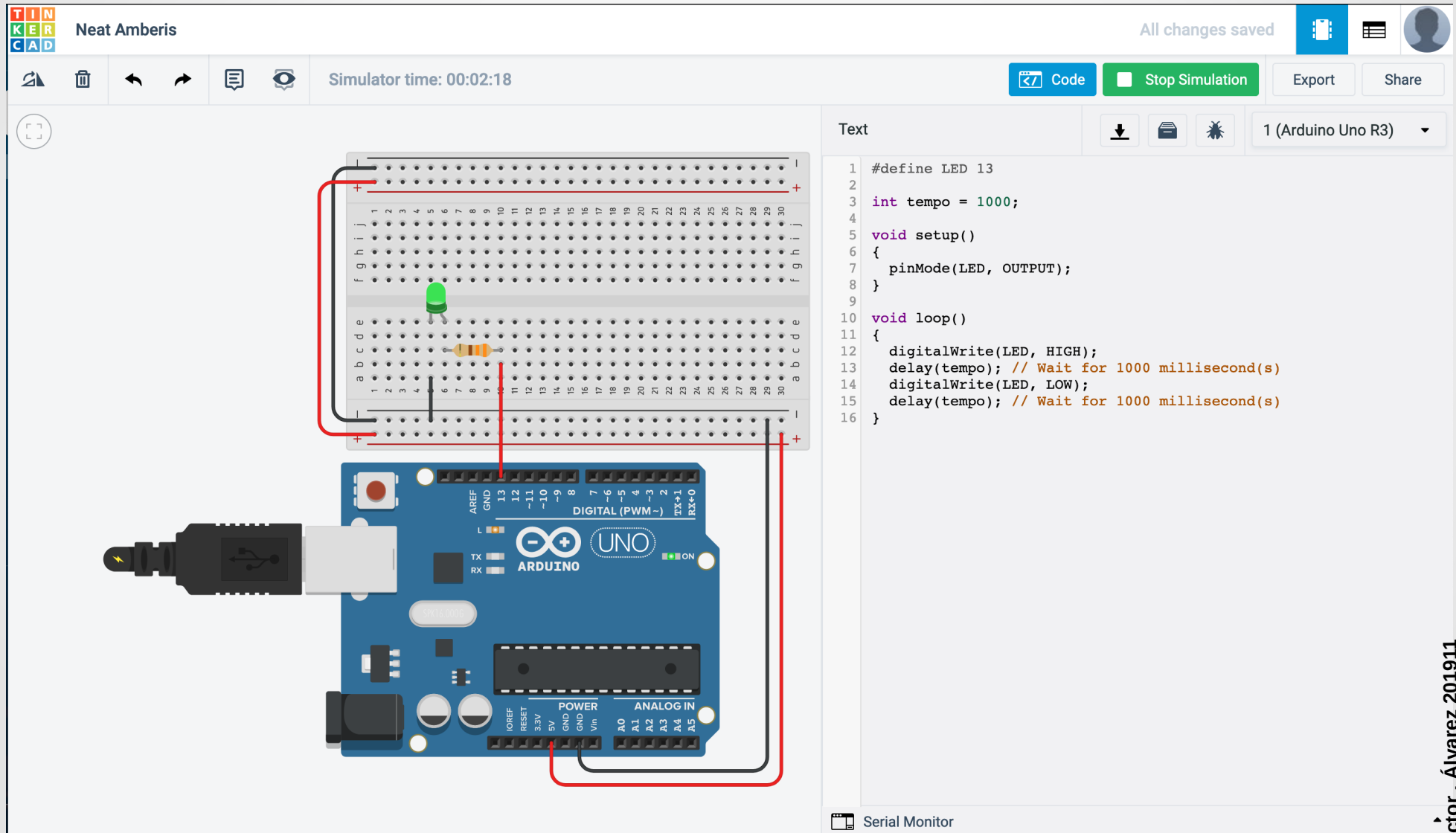
Neat Amberis

All changes saved

Simulator time: 00:02:18

Code Stop Simulation Export Share

1 (Arduino Uno R3)



```
1 #define LED 13
2
3 int tempo = 1000;
4
5 void setup()
6 {
7   pinMode(LED, OUTPUT);
8 }
9
10 void loop()
11 {
12   digitalWrite(LED, HIGH);
13   delay(tempo); // Wait for 1000 millisecond(s)
14   digitalWrite(LED, LOW);
15   delay(tempo); // Wait for 1000 millisecond(s)
16 }
```

Serial Monitor

Saídas dixitais – Encender varios LEDs

- Conecta cinco LEDs na placa de entrenamiento, coas súas correspondentes resistencias e prográmao con Sketch para que se encendan sucesivamente cun intervalo constante entre eles.
- Se o programa que fixeches escalase a máis LEDs ou actuadores, é fácil de manter?
- Que modificacións se poden facer para mellorar o programa en termos de:
 - Lexibilidade
 - Facer modificacións para incluír o dobre de LEDs
 - Facer modificacións por mantementos (outras demandas de tempos de espera ou mudar para pins dixitais diferentes)



Saídas digitais – Encender varios LEDs

```
cinco.leds.raw | Arduino 1.8.10

/*
 * Encende e apaga sucesivamente 5 LEDs
 * con frecuencia constante
 */
int tempo = 500;

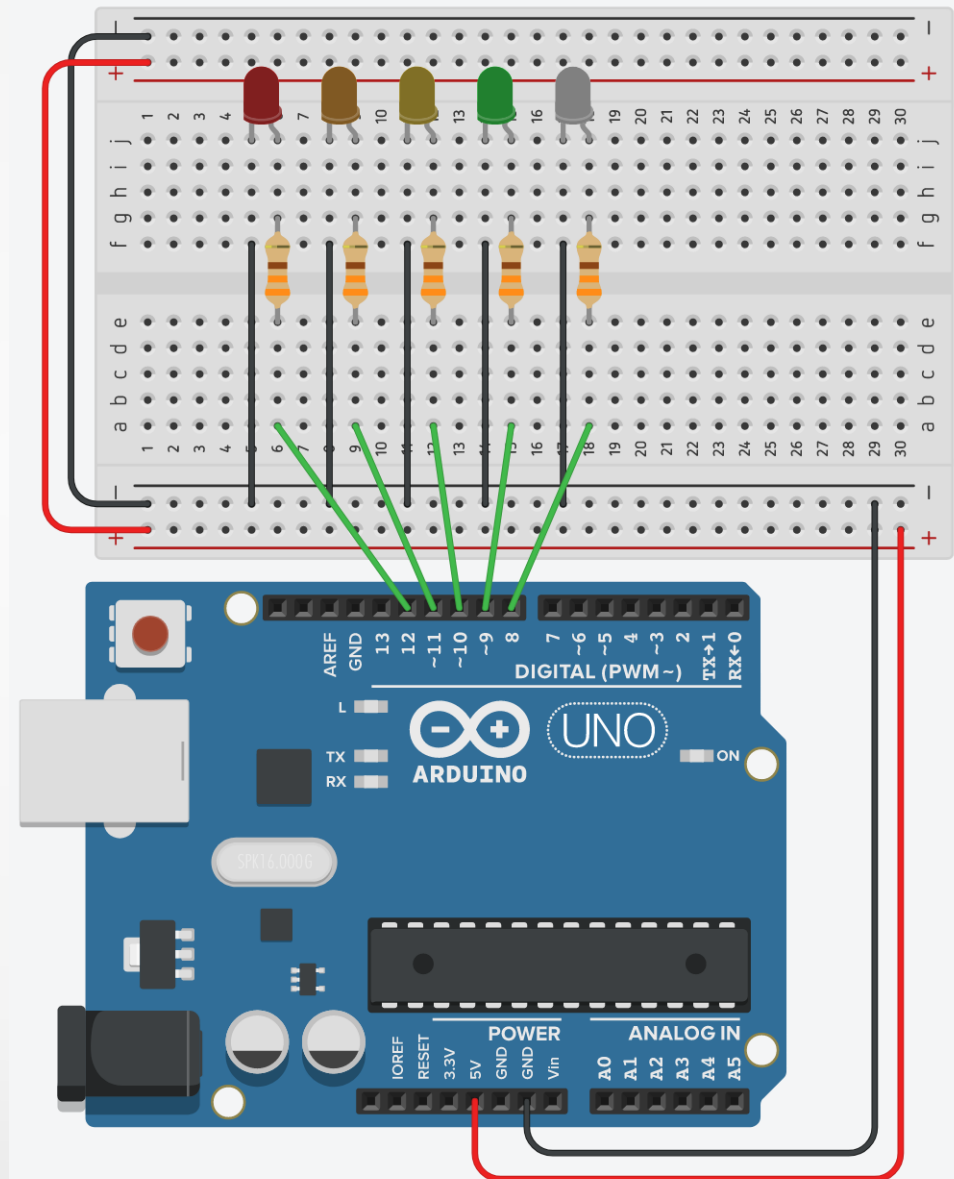
void setup() {
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop() {
  for(int i=8; i<13; i++) {
    digitalWrite(i, HIGH);
    delay(tempo);
    digitalWrite(i, LOW);
    delay(tempo);
  }
}
```

Compilação Terminada

0 rascunho usa 994 bytes (3%) do espaço de armazenamento
Variáveis globais usam 9 bytes (0%) de memória dinâmica

1 Arduino/Genuino Uno em /dev/cu.wchusbserial1410



Saídas digitais – Encender varios LEDs

```
cinco.leds.raw | Arduino 1.8.10

/*
 * Encende e apaga sucesivamente 5 LEDs
 * con frecuencia constante
 */
int tempo = 500;

void setup() {
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop() {
  for(int i=8; i<13; i++) {
    digitalWrite(i, HIGH);
    delay(tempo);
    digitalWrite(i, LOW);
    delay(tempo);
  }
}
```

Compilação Terminada

0 rascunho usa 994 bytes (3%) do espaço de armazenamento
Variáveis globais usam 9 bytes (0%) de memória dinâmica

1 Arduino/Genuino Uno em /dev/cu.wchusbserial1410

```
cinco.leds | Arduino 1.8.10

/*
 * Encende e apaga sucesivamente 5 LEDs
 * con frecuencia constante
 */
#define LED 8

int tempo = 500;

void setup() {
  //Inicializa 5 LEDs
  for(int i=0; i<5; i++) {
    pinMode(LED + i, OUTPUT);
  }
}

void loop() {
  //On/Off LEDs alternativamente
  for(int i=0; i<5; i++) {
    digitalWrite(LED + i, HIGH);
    delay(tempo);
    digitalWrite(LED + i, LOW);
    delay(tempo);
  }
}
```

Compilação Terminada

0 rascunho usa 974 bytes (3%) do espaço de armazenamento
Variáveis globais usam 9 bytes (0%) de memória dinâmica

17 Arduino/Genuino Uno em /dev/cu.wchusbserial1410

Saídas digitais – Encender varios LEDs

- Realiza un programa que haga o mesmo percorrido de encender e apagar sucesivamente os LEDs, mais empregando en exclusiva bucles while.
- A estrutura dun bucle while é como sigue:

```
// Estructura dun bucle while

//Declaración de variable de control
//Condición de entrada no bucle
// Dentro do bucle:
//   - Tarefas repetitivas
//   - Actualización da variable de control
int i = 0;
while(i < condicion) {
    // Tarefas para repetir
    i++;
}
```

Saídas dixitais – Encender varios LEDs

- O seguinte 'script' mostra dúas estruturas iterativas, while() e for().
- Convén recordar que os bucles for() só se empregan cando sabemos exactamente cantas iteracións debemos facer.

```
// Estructura dun bucle while

//Declaración de variable de control
//Condición de entrada no bucle
// Dentro do bucle:
//   - Tarefas repetitivas
//   - Actualización da variable de control
int i = 0;
while(i < condicion) {
    // Tarefas para repetir
    i++;
}
```

```
//Estructura dun bucle for

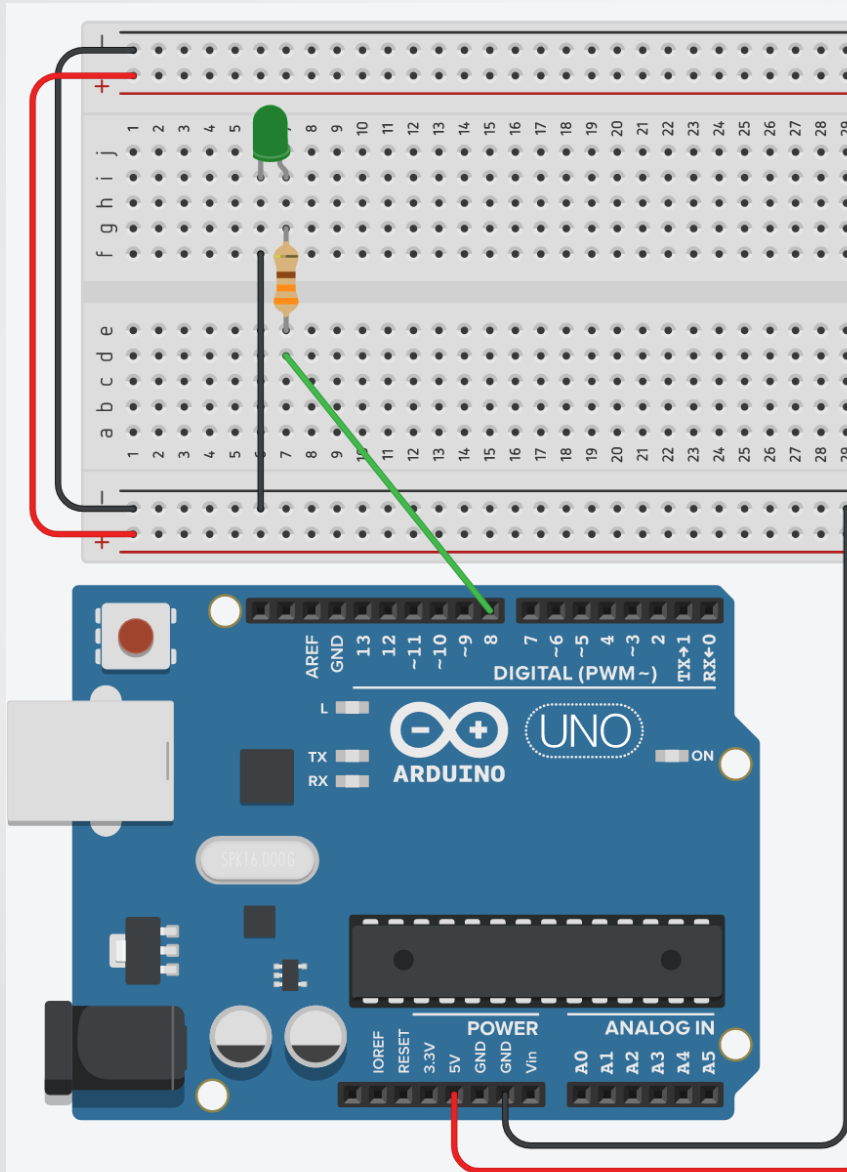
//Temos que saber o número de veces que se
//repite a iteración
for(int i = 0; i < condicion; i++) {
    //Tarefas para repetir
}
```


Saídas dixitais – Imitar vela cun LED

- Aproveita un dos LEDs do exercicio anterior, na placa de entrenamiento, para encendelo e apagalo con frecuencia variable, de xeito que se asemelle a unha vela.
- Non imos emular polo de agora a intensidade variable da vela.
- Pódese reaproveitar parte do código do exercicio anterior?
- Que modificacións se poden facer para mellorar o programa en termos de:
 - Lexibilidade
 - Mantemento posterior
 - Pódese escalar o programa para facelo con dous LEDs?

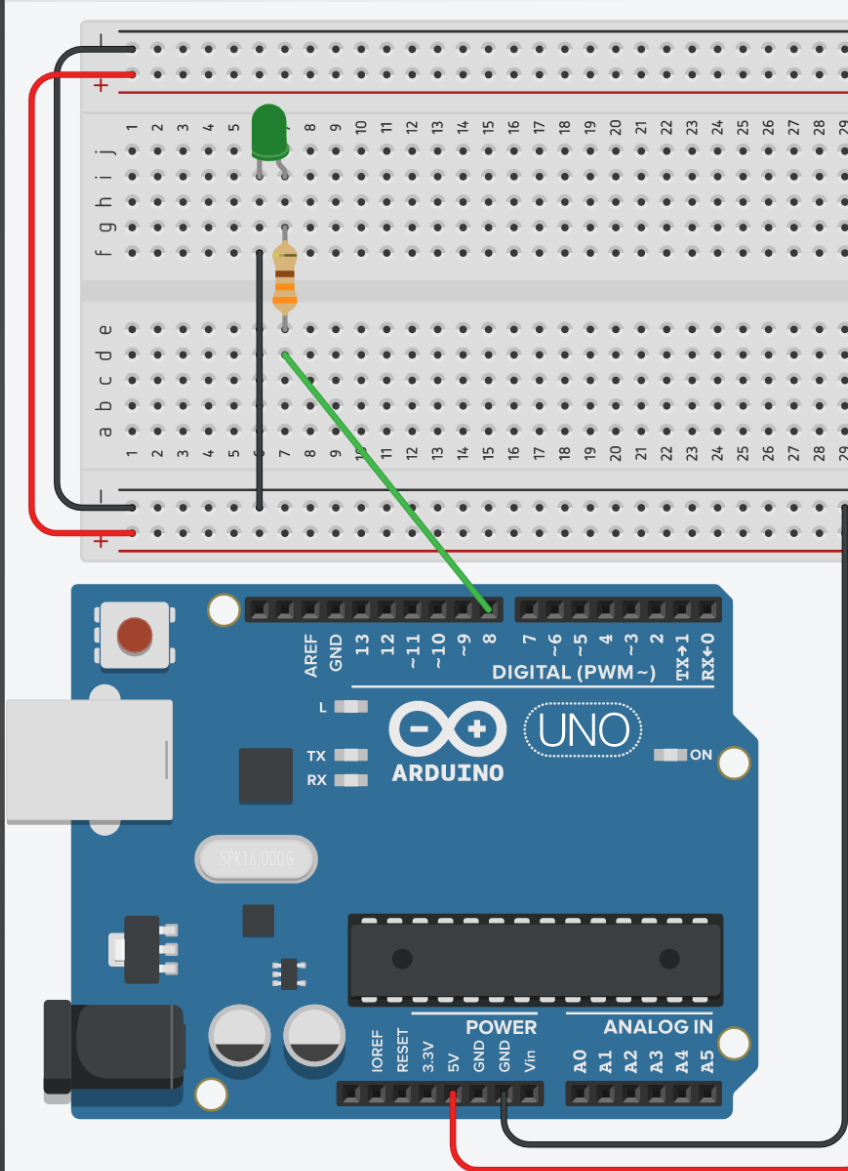


Saídas digitais – Imitar vela cun LED



```
1
2 #define VERDE 8
3
4 long tempo;
5 int tmin = 200, tmax = 1500;
6
7 void setup()
8 {
9   pinMode(VERDE, OUTPUT);
10  randomSeed(23);
11 }
12
13 void loop() {
14   digitalWrite(VERDE, HIGH);
15   tempo = random(tmin, tmax);
16   delay(tempo);
17   digitalWrite(VERDE, LOW);
18   tempo = random(tmin, tmax);
19   delay(tempo);
20 }
21
```

Saídas digitais – Imitar vela cun LED



```
1
2 #define VERDE 8
3
4 long tempo;
5 int tmin = 200, tmax = 1500;
6
7 void setup()
8 {
9     pinMode(VERDE, OUTPUT);
10    randomSeed(23);
11 }
12
13 void loop() {
14     digitalWrite(VERDE, HIGH);
15     tempo = random(tmin, tmax);
16     delay(tempo);
17     digitalWrite(VERDE, LOW);
18     tempo = random(tmin, tmax);
19     delay(tempo);
20 }
21
```

```
1
2 #define VERDE 8
3
4 void setup()
5 {
6     pinMode(VERDE, OUTPUT);
7     randomSeed(23);
8 }
9
10 void loop() {
11     vela(VERDE);
12 }
13
14 void vela(int led) {
15     long tempo;
16     int tmin = 200, tmax = 1500;
17     digitalWrite(led, HIGH);
18     tempo = random(tmin, tmax);
19     delay(tempo);
20     digitalWrite(led, LOW);
21     tempo = random(tmin, tmax);
22     delay(tempo);
23 }
```

Saídas dixitais

- Nesta unidade aprendemos a:
 - crear, editar, compilar e cargar un script
 - declarar un pin como saída dixital -pinMode()- e escribir nel un valor HIGH ou LOW -digitalWrite()-
 - declarar parámetros e variables
 - usar estruturas iterativas básicas -while() e for()-
 - escoller números aleatorios -randomSeed() e random()-
 - declarar unha función definida por nós mesmos e usala posteriormente