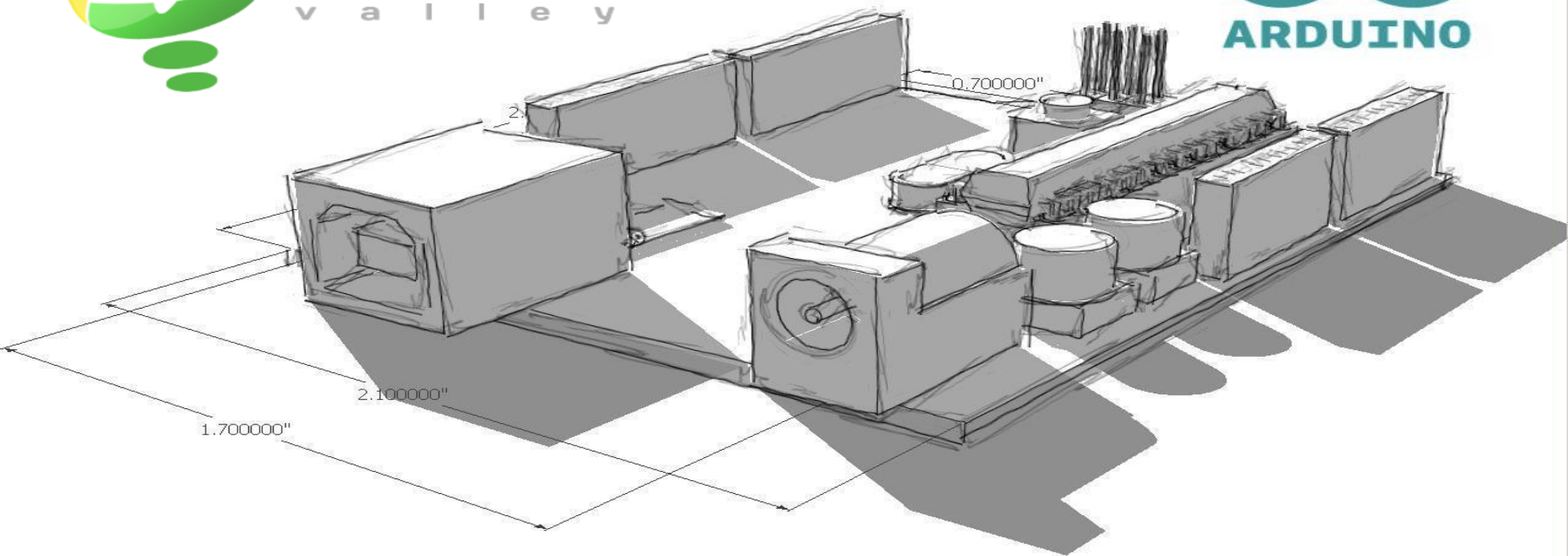




**CyLicon**  
v a l l e y



# Arduino - Semana 3

## 10 de Mayo 2014

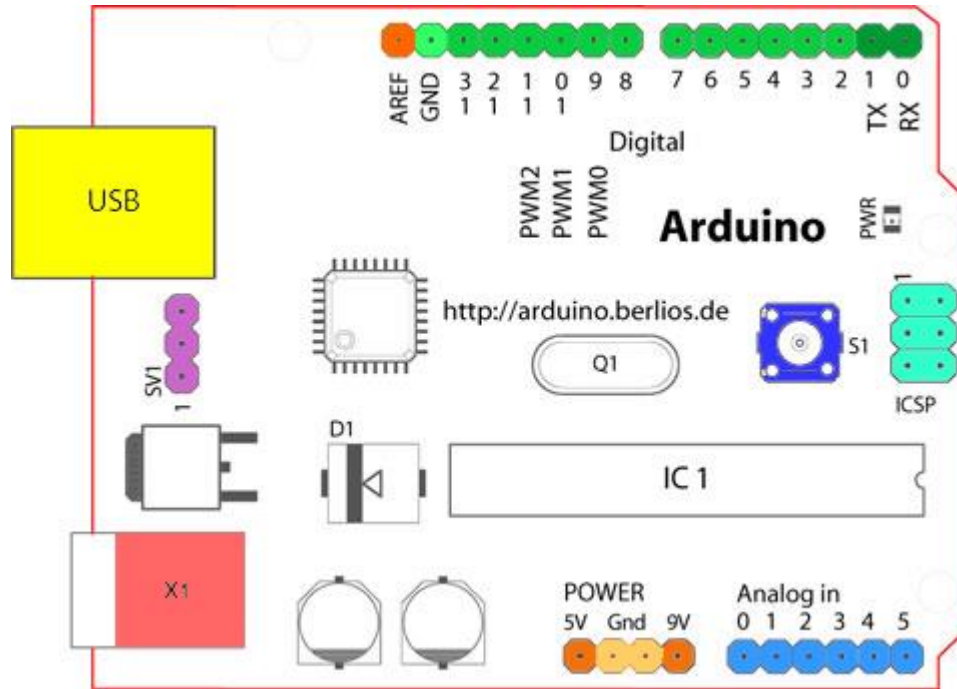


# Agenda Tercera Sesión

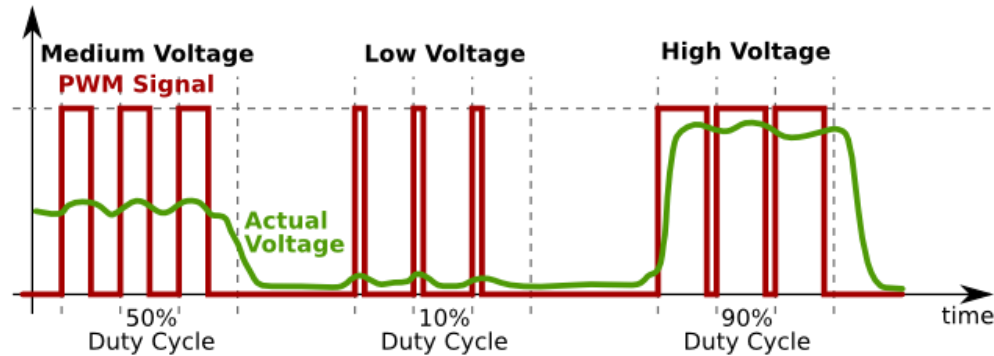
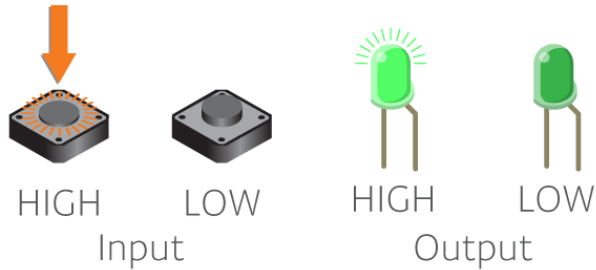


- Digital vs. Analógico.
- Speakers
- Sensores capacitivos
- Preparación para la sesión 4.

# Placa Arduino Uno



# Digital vs. analógico



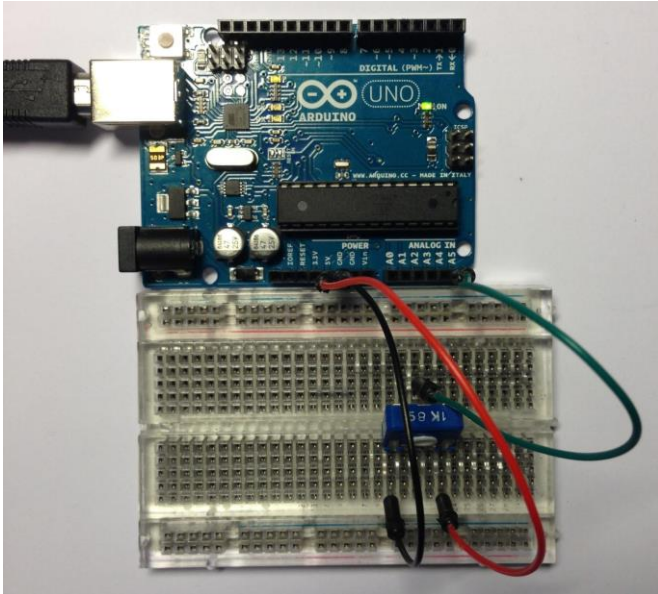
# Potenciómetro



- Resistencia variable dependiendo de la posición.
- Distintos tipos.  
Según uso: rotatorio, deslizante, múltiple.  
Según respuesta: lineal, logarítmica, antilogarítmico...

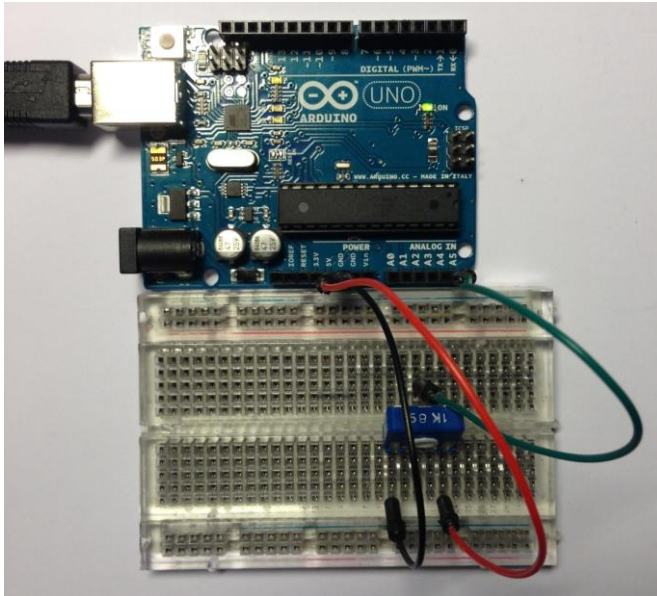


# Leyendo en analógico I



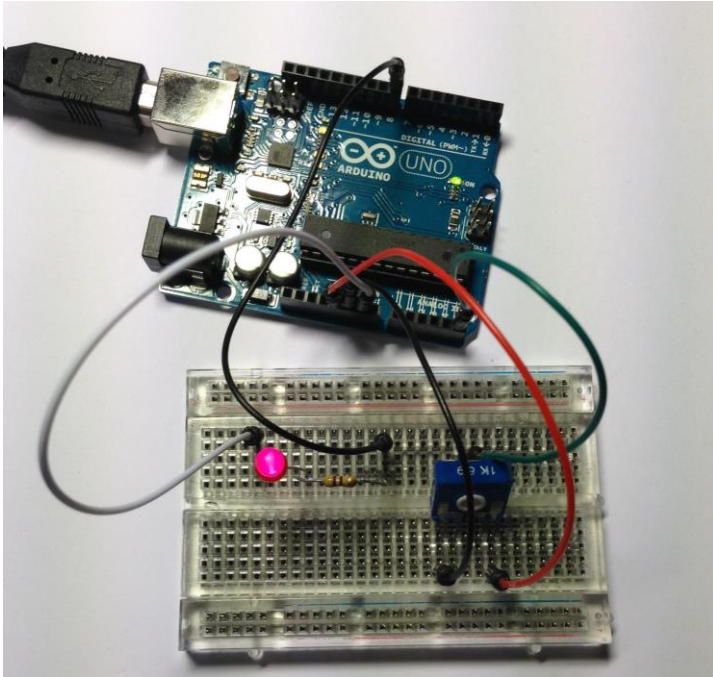
- Valores de 0 a 1024.
- Pines analógicos sólo funcionan como entradas.
- Utilización de la función `map()`.
- `val = map(val, 0, 1023, 0, 255);`  
`val = map(val, 0, 1023, 0, 1); //`  
`digital`

# Leyendo en analógico II



```
1 void setup() {  
2     Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6     int val = analogRead(A5);  
7     delay(val);  
8     Serial.println(val);  
9 }
```

# Leyendo en analógico III



```
1  int ledPin = 8;
2
3  void setup() {
4      pinMode(ledPin, OUTPUT);
5  }
6
7  void loop() {
8      int val = analogRead(A5);
9      digitalWrite(ledPin, HIGH);
10     delay(val);
11     digitalWrite(ledPin, LOW);
12     delay(val);
13 }
```



# Escritura analógica

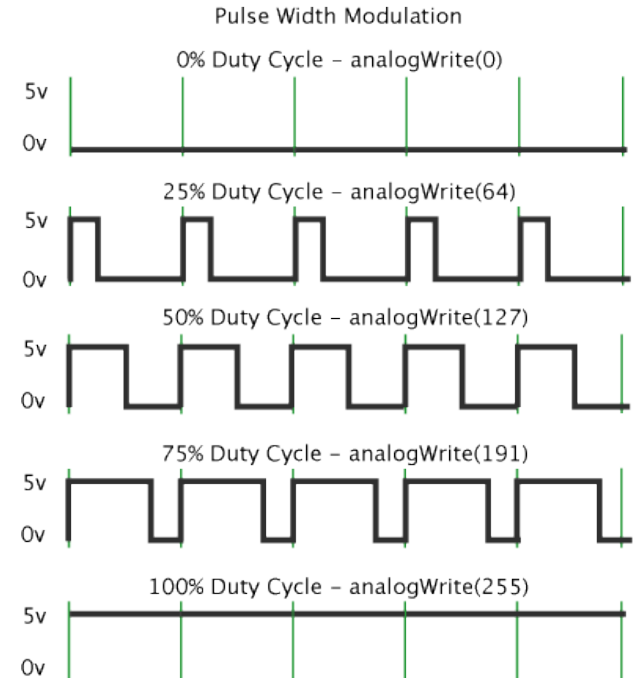


- En la semana 2 vimos cómo leer valores analógicos (*analogRead()*).
- Vamos a utilizarlo para escribir un valor analógico.

# Ejercicio 2 - Led difuminado



- Para difuminar un Led necesitamos un pin con escritura analógica activada: Pulse-Width Modulation ([PWM](#)).
- Están marcados en el board (~).

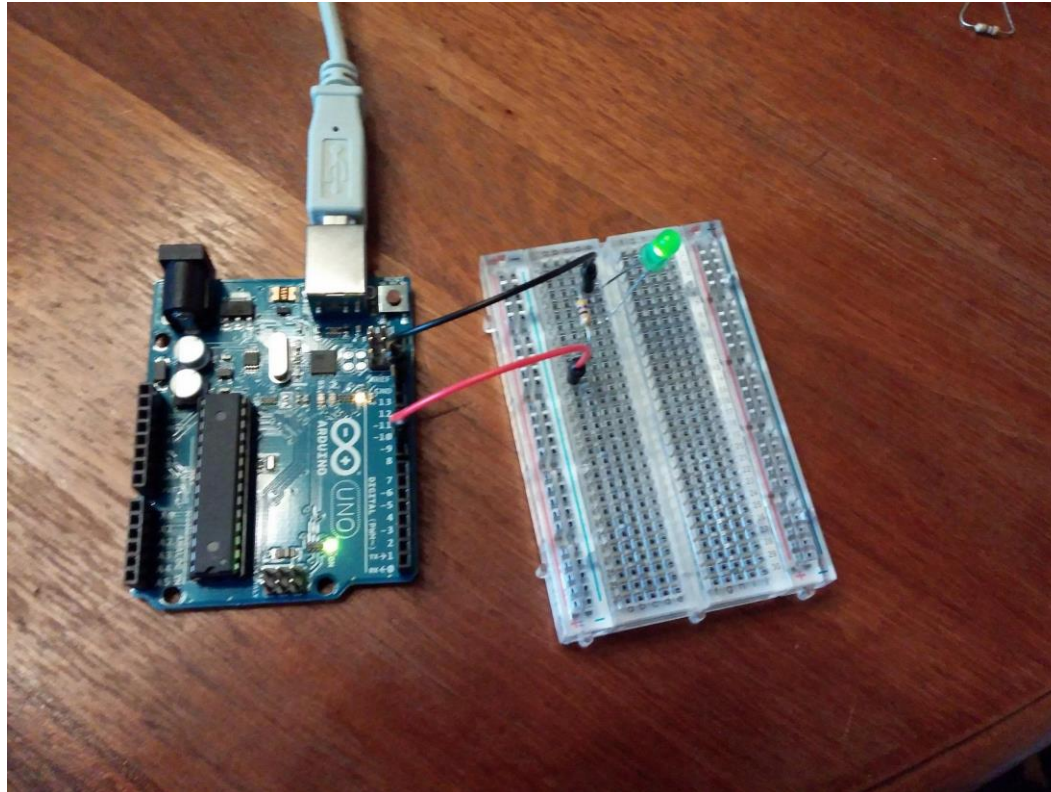


# Ejercicio 2 - Led difuminado



- Utilizad *analogWrite* para iluminar un led poco a poco.
  - Probad con el output 9 y con el 12.
  - Código

# Ejercicio 2 - Led difuminado



# Ejercicio 2 - Led difuminado

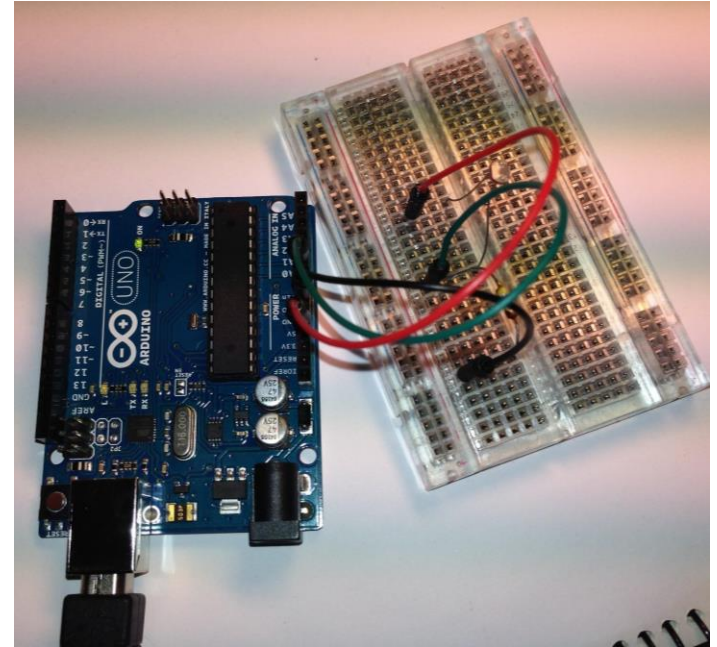


```
1  int led = 9;
2  int brillo = 0;
3
4  void setup() {
5
6  }
7
8  void loop() {
9      for(brillo = 0 ; brillo <= 5000; brillo +=50) {
10         analogWrite(led, brillo);
11         delay(300);
12     }
13
14     if (brillo >= 5000) {
15         brillo = 0;
16     }
17 }
```

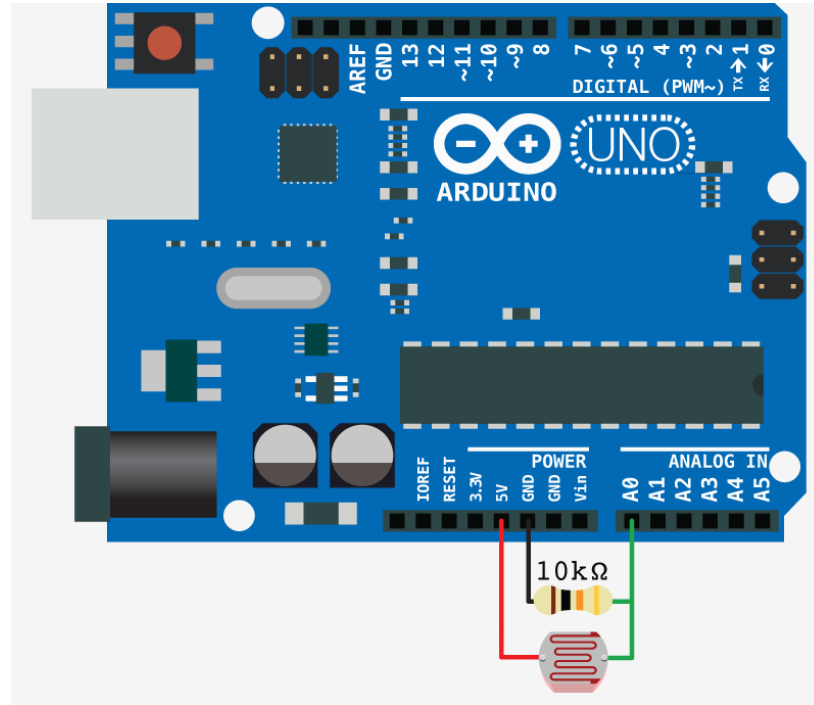
# Input analógico: LDR



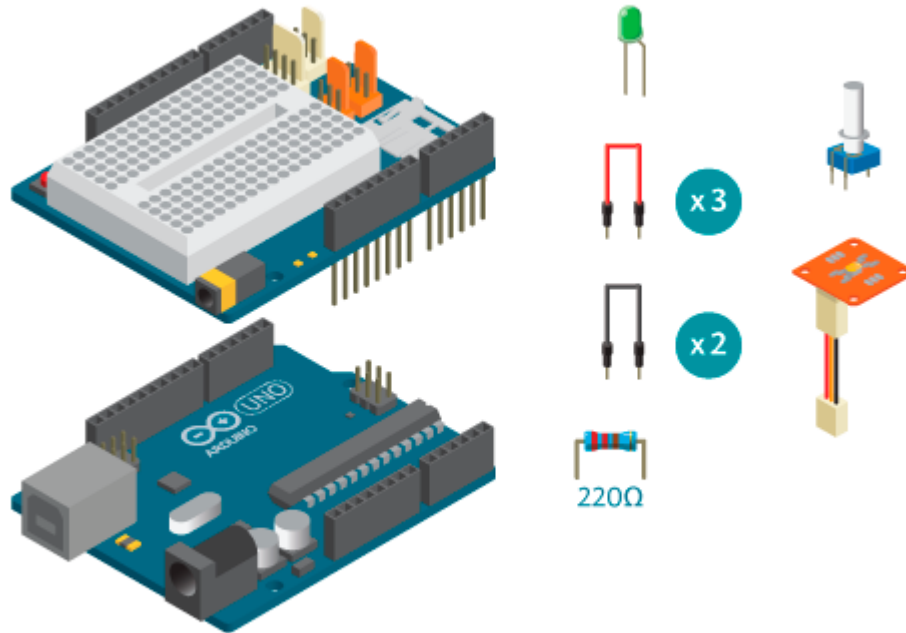
- En el kit utilizado, viene un sensor de luz y uno de temperatura.
- Hace falta calibrarlos.



# Input analógico: LDR



# Input analógico: calibración





# Output analógico: Speakers



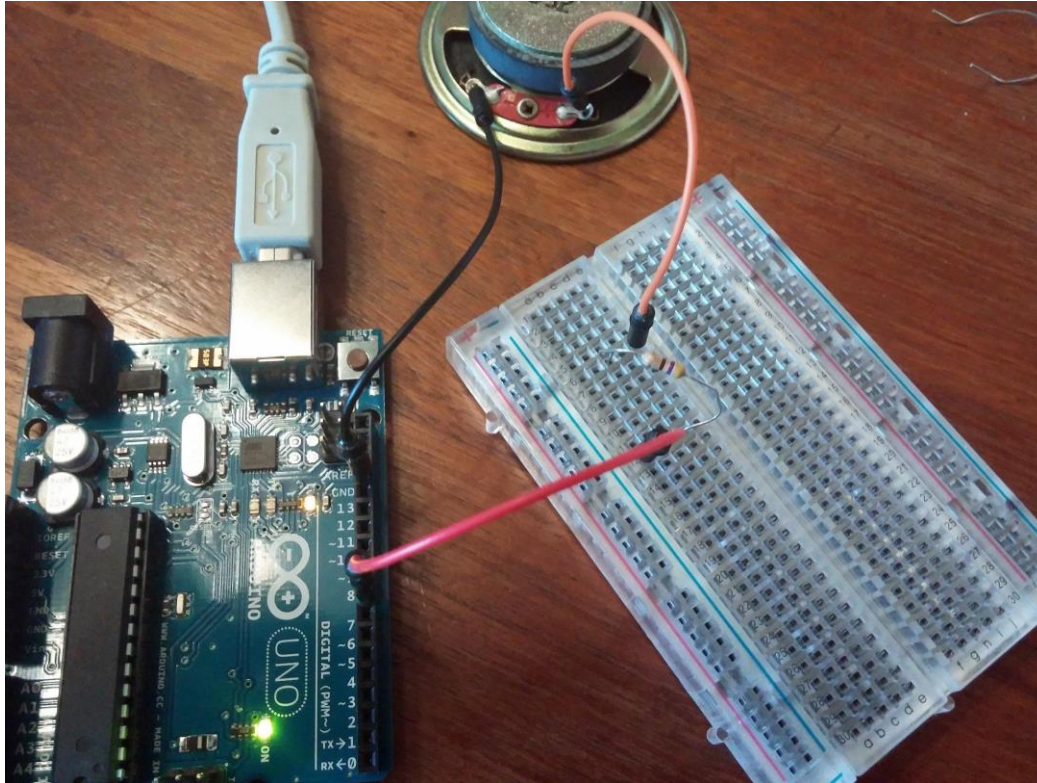
- Vamos a jugar con altavoces de 8 ohmios.
- Por seguridad **no deberíamos conectarlo directamente** (podemos volar una salida).
- Deberíamos usar una resistencia de 100 ohm.

# Ejercicio 2 - Speakers básicos



- Montad un altavoz
  - Usad la función *tone(pin, valor, tiempo)*.
  - Que reproduzca el sonido 262.
  - Probad a usar una resistencia mayor que 4xx ohm.
  - [Código](#)

# Ejercicio 2 - Speakers básicos



```
1 void setup() {  
2     tone(8, 254, 500);  
3 }  
4  
5 void loop() {  
6  
7 }
```

# Speakers básicos

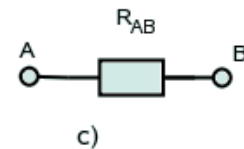
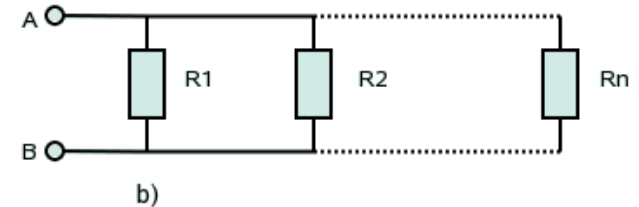
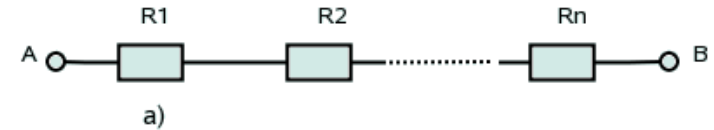


- El altavoz suena poco:
  - El típico circuito necesita un amplificador (con un capacitador/condensador).
  - Estamos usando una resistencia mayor.

# Speakers básicos



- Podríamos combinar resistencias para obtener una más adecuada:
  - Resistencias en serie  
 $R_t = R_1 + R_2$
  - Resistencias en paralelo  
 $1/R_t = 1/R_1 + 1/R_2$



# Speakers básicos



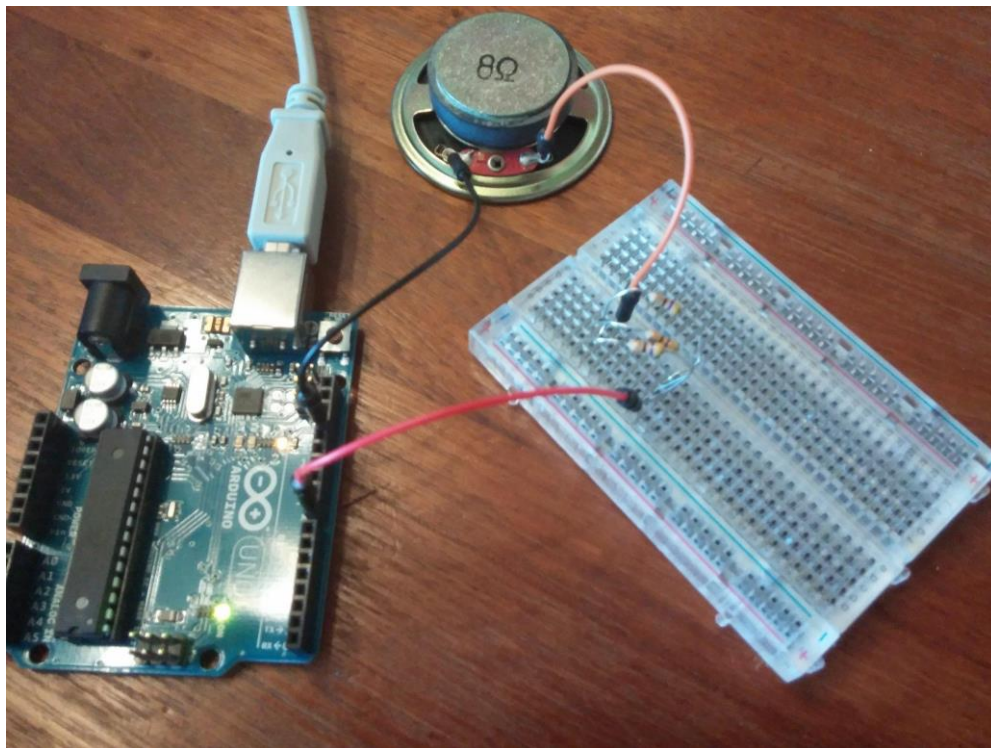
- Para el mismo valor de las resistencias, en paralelo:
  - $1/R_t = 1/R_1 + 1/R_1 = 2/R_1 \rightarrow R_t = R_1/2.$

# Speakers básicos



- Utilizad resistencias en paralelo (sin bajar de 100 ohm) para aumentar el sonido.

# Speakers básicos





# Ejercicio 2c - Melodía



Ahora que tenemos sonido podemos tocar un

- [Notas](#)
- [Código](#)
- Funciona bien esta secuencia: NOTE\_C  
NOTE\_G3, NOTE\_G3, NOTE\_A3, NOTE\_C  
NOTE\_B3, NOTE\_C4

executable file | 96 lines (92 sloc) | 2

```
1  /*****
2  * Public Constants
3  *****/
4
5  #define NOTE_B0  31
6  #define NOTE_C1  33
7  #define NOTE_CS1 35
8  #define NOTE_D1  37
9  #define NOTE_DS1 39
10 #define NOTE_E1  41
11 #define NOTE_F1  44
12 #define NOTE_FS1 46
13 #define NOTE_G1  49
14 #define NOTE_GS1 52
15 #define NOTE_A1  55
16 #define NOTE_AS1 58
17 #define NOTE_B1  62
18 #define NOTE_C2  65
19 #define NOTE_CS2 69
20 #define NOTE_D2  73
21 #define NOTE_DS2 78
22 #define NOTE_E2  82
23 #define NOTE_F2  87
```

## Ejercicio 2d - Iterar valores



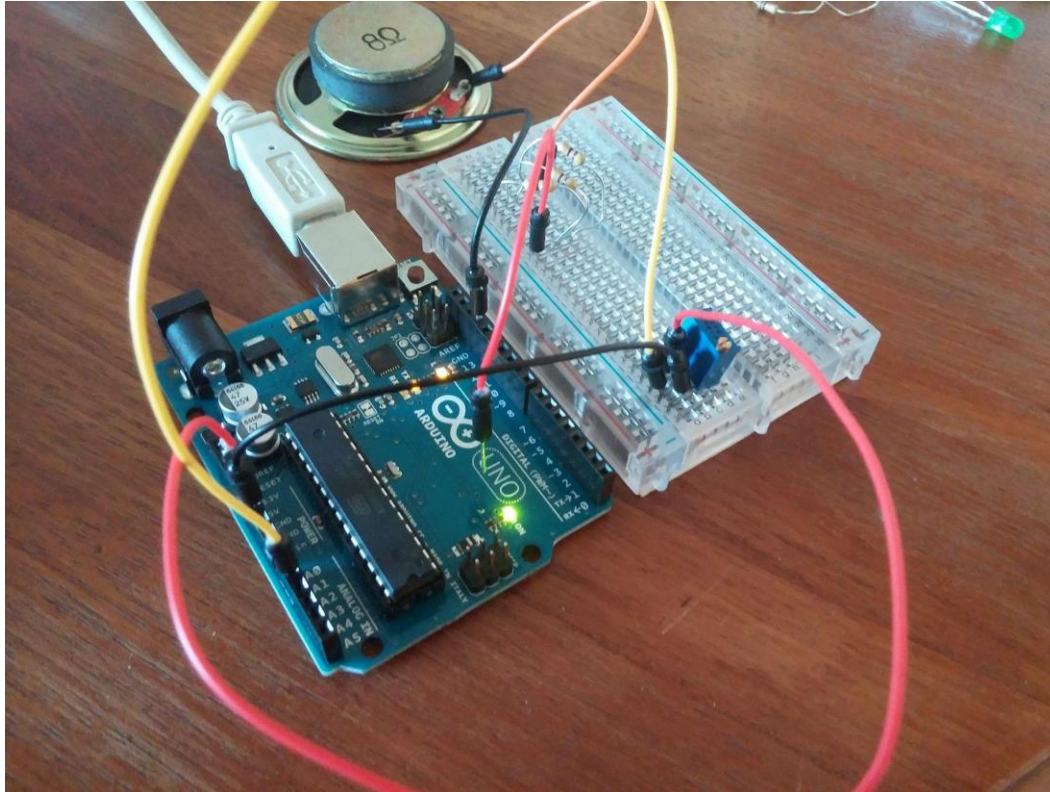
También podemos iterar sobre los valores de las notas y reproducir todas.

## Ejercicio 2e



- O usar el potenciómetro para controlar la nota ejecutada.
  - [Código](#)

# Ejercicio 2e



# Sensores capacitivos



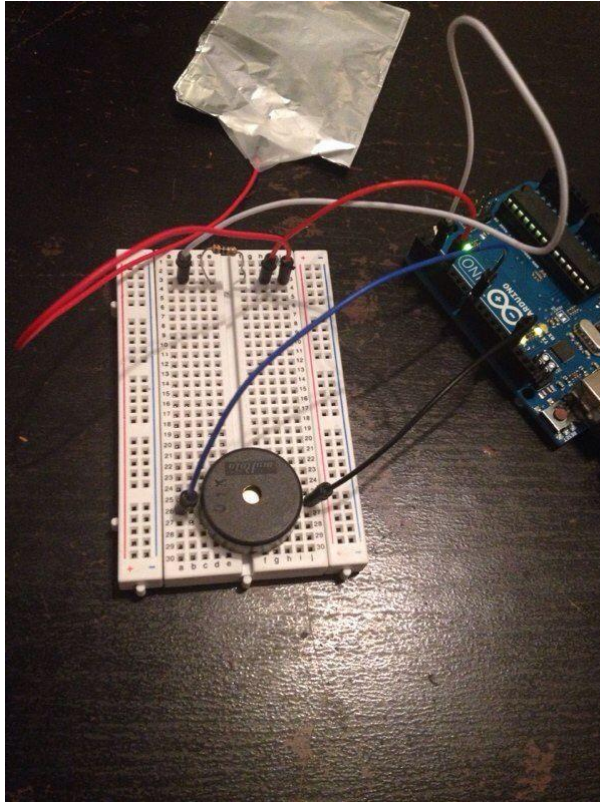
El papel albal nos puede servir de interruptor combinado con una resistencia

# Ejercicio 3



- Usad papel albal para controlar un speaker o un led.
- Necesita una resistencia de 1M.

# Ejercicio 3



# POV



- Persistence of Vision (POV)
- O la prueba local

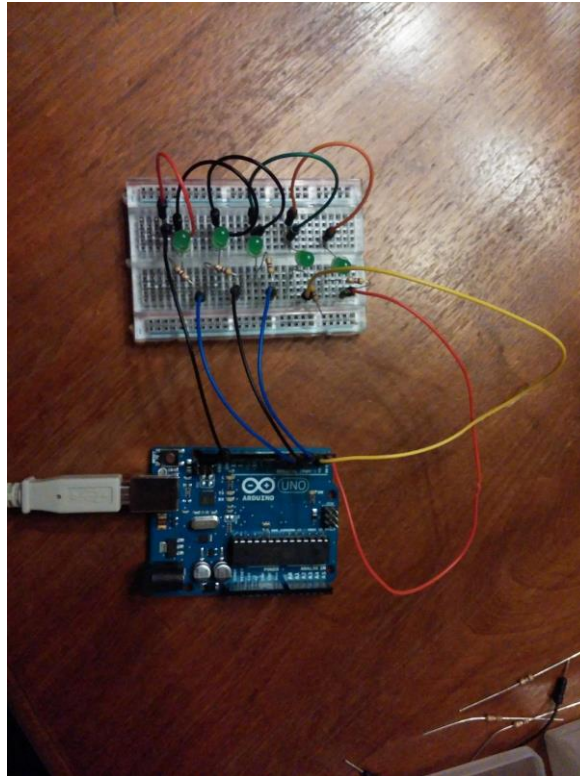


# Ejercicio 4



- Montad una pequeña prueba de POV
  - 5 leds en fila, con resistencia.
  - [Código](#)

# Ejercicio 4



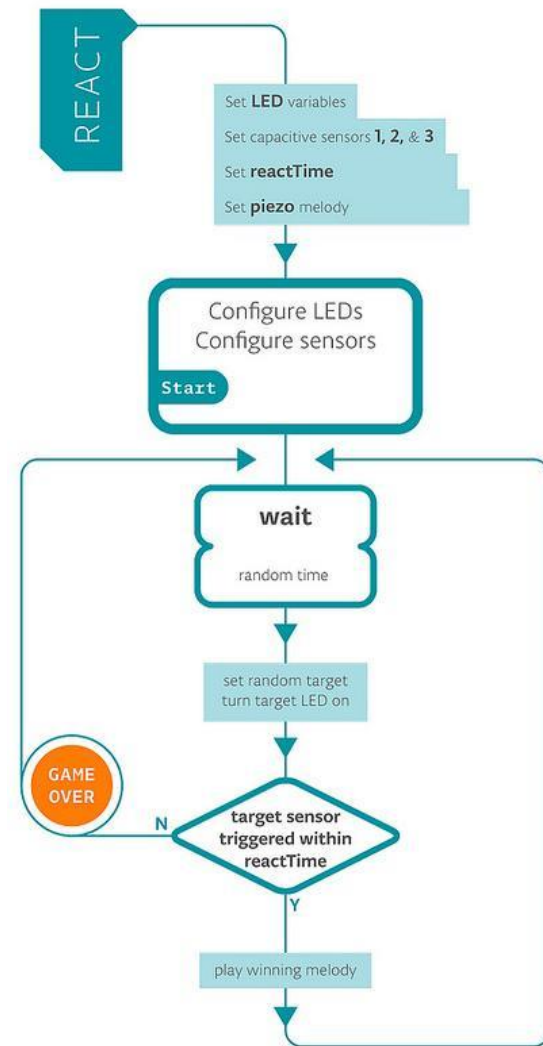
# Ejercicio 5



- Nos complicamos un poco:
  - Simon says
  - 2/3/4 leds (con resistencia) y 2/3/4 pulsadores
  - random()
  - video [[casero](#)][[matrices](#)][[olímpico](#)]

# Ejercicio adicional

- Reacciona:
  - Uno de los tres LEDs se iluminarán aleatoriamente.
  - Tienes que golpear el sensor capacitivo correspondiente dentro del tiempo de reacción



# Ejercicios adicionales Semana 3



- Reproductor de sonidos (altavoz + tarjeta SD).
- Monstruo de las galletas (webcam).
- Drawdio (lápiz y papel).
- Caja knock knock (piezoeléctrico como sensor).

# Semana 4



- Motores.
- Servos.
- Robots!!!

A word cloud featuring the phrase "Thank you" in numerous languages. The word "Gracias" is the largest and most central. Other prominent words include "Merci", "Hvala", "Ngiyabonga", and "Dankon". The words are arranged in a circular pattern around these central terms, with many smaller words in various colors and orientations.