



# Camaleón Electrónico Iluminado

**Rafael Díaz Robles**  
**Juan Manuel Aragón Galán**  
I.E.S Politécnico Jesús Marín  
S21ME - 2019/2020

## **Índice**

<b>1.Título</b>	<b>2</b>
<b>2.Participantes</b>	<b>2</b>
<b>3.Descripción</b>	<b>2</b>
<b>4.Materiales</b>	<b>2</b>
<b>5.Electrónica</b>	<b>3</b>
<b>6.Diagrama de Flujo</b>	<b>7</b>
<b>7.Códigos de programas</b>	<b>8</b>
<b>8.Imágenes del Proyecto</b>	<b>16</b>
<b>9.Incidencias</b>	<b>16</b>
<b>10.Web utilizadas</b>	<b>18</b>

*"En este enlace a drive se podrá descargar todo el material del proyecto (códigos, fotos, y proyecto en pdf y word)."*

[https://drive.google.com/open?id=1Zg1P5V2qkgCQYHVm\\_f3n3ciuHleDDmTD](https://drive.google.com/open?id=1Zg1P5V2qkgCQYHVm_f3n3ciuHleDDmTD)

## 1. Título

Camaleón electrónico iluminado

## 2. Participantes

-Rafael Díaz Robles

-Juan Manuel Aragón Galán

## 3. Descripción

Principalmente haremos el proyecto en una maqueta basándonos en el sensor de color TCS3200 y leds rgb y haremos también otra pequeña maqueta simulando el ojo de un camaleón. Posteriormente, para subir nota, haremos el camaleón grande con el sensor de color y el movimiento de los ojos.

Este (el camaleón grande) es un trabajo conjunto con el grupo de arte de bachillerato. Vamos a realizar un proyecto que basará en un camaleón gigante, el cual se iluminará del color que detecte a través de uno de sus sensores y moverá los ojos mediante el uso de servos independientes, haciendo el efecto de la vista del animal.

## 4. Materiales

<u>Ojo</u>	<u>Camaleón</u>
-Madera	-Baúl de madera
-Pegamento	-Diseño 3D Camaleón
-Tornillos, arandelas y tuercas	-Termoretractil
-Tapón de botella	-Cartulina de colores
-Tapadera de un spray (Para hacer el círculo negro del ojo)	-Chapón de madera
-Cables y placa protoboard	-Cristal de metacrilato
-2xServo SG90	-Cables y placa protoboard
-1xArduino Uno	-1xSensor TCS3200
	-1xArduino UNO

## 5. Electrónica

### Maqueta Ojo

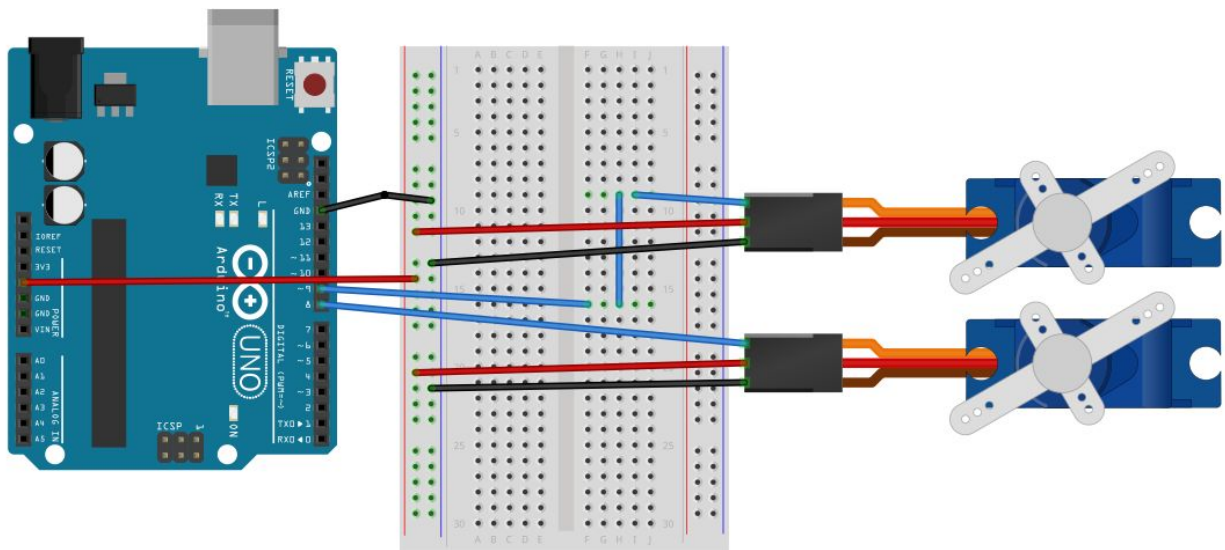
#### Motor1

- Cable Naranja → Pin 9 Arduino
- Cable Marrón → GND
- Cable Rojo → 5v

#### Motor2

- Cable Naranja → Pin 8 Arduino
- Cable Marrón → GND
- Cable Rojo → 5v

#### Esquema de conexión



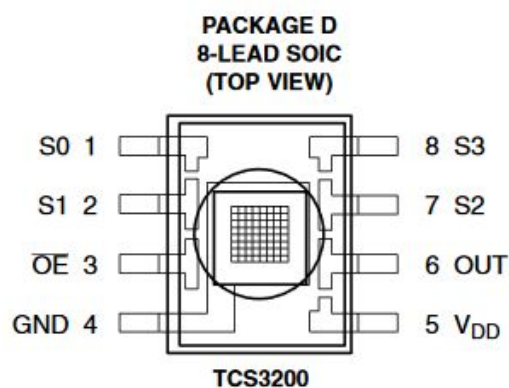
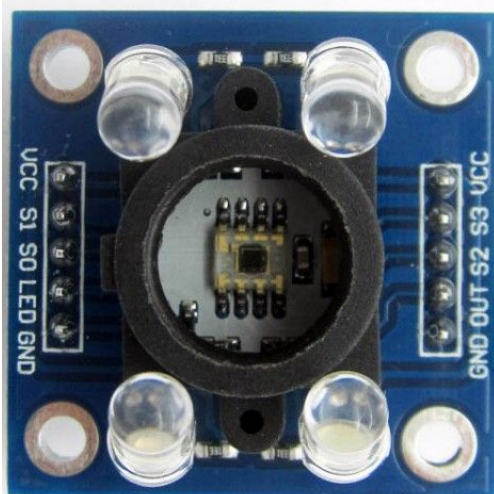
Su funcionamiento será sencillo, simplemente se moverán ambos servos de 0 a 120° y de 120° a 0°. Simulará el movimiento del ojo de un camaleón.

## Maqueta Camaleón

### Sensor TCS3200

El sensor de color TCS3200 es un dispositivo electrónico que nos permite interpretar el color de una manera equivalente a la del ser humano. El principio básico es que este sensor dispone de unos fotorreceptores a modo de matriz que nos permite identificar los colores Rojo - Verde - Azul.

Dentro de esta cavidad podemos observar un elemento con 8 patillas. Ese elemento es el fotorreceptor capaz de proporcionar a nuestro robot el color sobre el que está apuntando. Alrededor del mismo se disponen 4 LEDs que proporcionan luz a la superficie sobre la que estamos leyendo el color.



Como podemos observar disponemos las 8 patillas del SOIC, no se corresponden exactamente con la imagen de la izquierda 10 ya que se repiten 2 GND y 2 VCC.

Los pines que nos interesan son con S0, S1, S2, S3, OUT, VCC y GND.

El sensor de color ha de proporcionarnos 3 valores para poder definir el color que está leyendo. La combinación de estos 3 colores, que son el Rojo, Verde y Azul para obtener el resto.

Por ejemplo, para obtener el color amarillo, será una combinación predominante de los colores Rojo y Verde y muy poco o nada de azul. Y así con todos.

El problema, es que los fotorreceptores no se activan todos a la vez, sino de uno en uno. Así que primero deberemos leer la cantidad de rojo, después medir la cantidad de verde, luego la de azul, y pasarlo a una escala determinada de colores para poder interpretarlo.

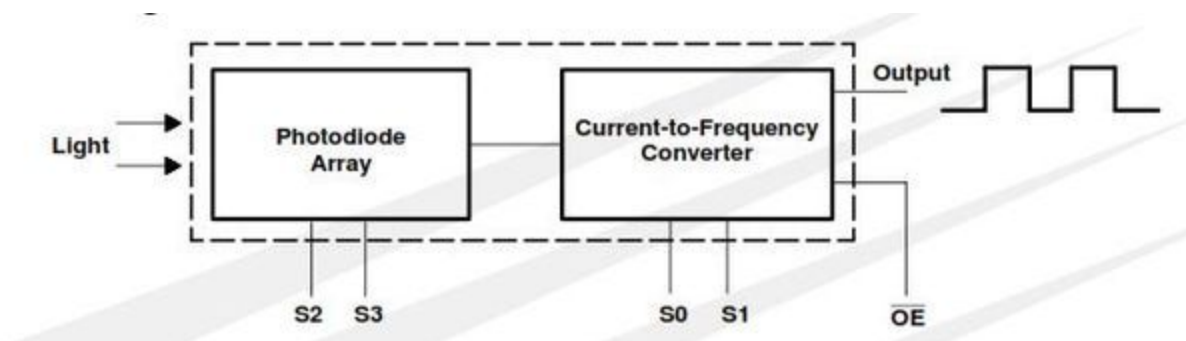
Para controlar estas lecturas, el fabricante, nos proporciona 2 pines de control que se asocian al **S2 y al S3** para saber cuál estamos midiendo y el pin de lectura, es el que se nos indica cómo **OUT**.

A través de la siguiente tabla podemos observar cómo activar estas lecturas.

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

La señal de salida que obtenemos es una señal cuadrada, que será más estrecha o más ancha en función del valor de rojo, verde o azul que estamos leyendo.

Como vemos en la siguiente imagen:



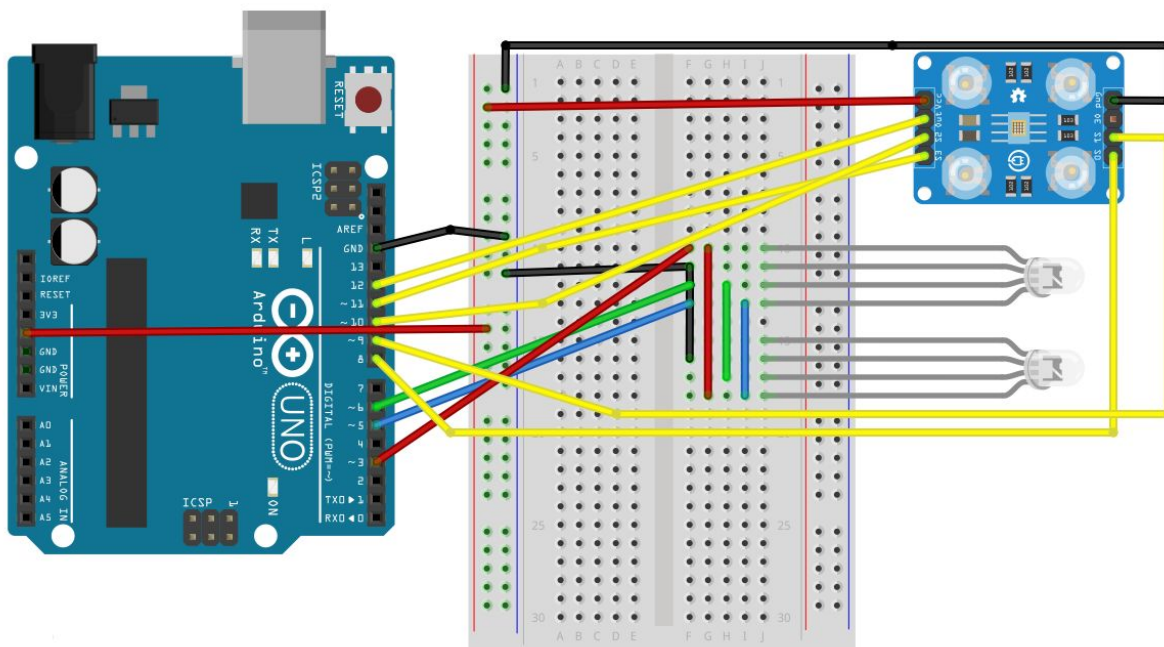
Las señales conectadas a los pines **S0 y S1**, son para aumentar el escalado de la señal. Cuanta más señal, más preciso se puede hacer el sensor de color. Aunque pueda gastar más energía. En la práctica esto nos permitirá reconocer una mayor gama de colores.

Nosotros lo dejamos como H - H  $\rightarrow$  100%.

Aquí vemos las distintas posibilidades:

S0	S1	OUTPUT FREQUENCY SCALING ( $f_o$ )
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

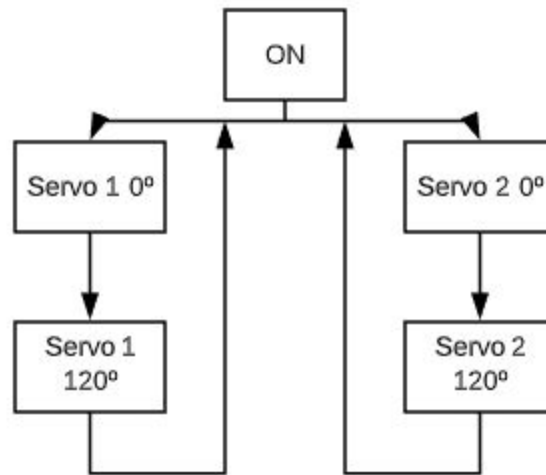
### Esquema de Conexión



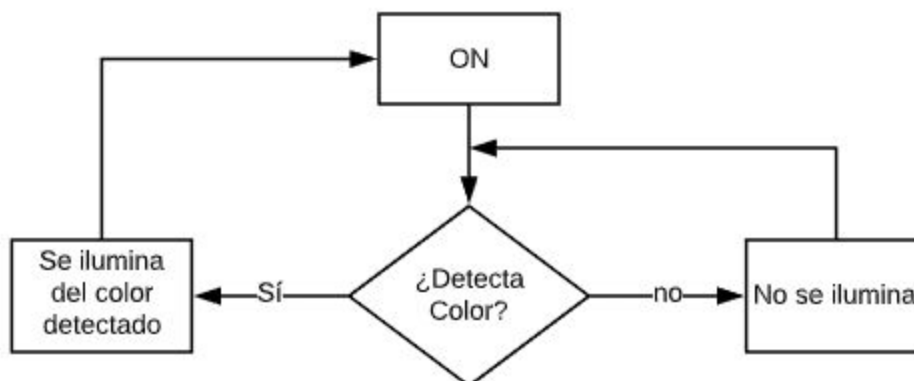
Su funcionamiento será que el camaleón en la parte de abajo tiene el sensor y cuando se ponga encima de un color determinado se iluminará de ese color.

## 6. Diagrama de Flujo

### Ojo



### Camaleón





## 7. Códigos de programas

### **Ojo**

```
// Incluimos la librería para poder controlar el servo
#include <Servo.h>

// PIN 9 Cable naranja
// PIN GND Cable marrón
// PIN 5v Cable rojo

// Declaramos la variable para controlar el servo
Servo servoMotor1;
Servo servoMotor2;

int pos=0;
// int pos2=0;

void setup() {
  // Iniciamos el monitor serie para mostrar el resultado
  Serial.begin(9600);

  // Iniciamos el servo para que empiece a trabajar con el pin 9
  servoMotor1.attach(9);
  servoMotor2.attach(8);
}

void loop() {
```

```

// Desplazamos a la posición 0°
for (pos = 0; pos <= 120; pos += 2) { //Movimiento de 0 a 120°
  servoMotor1.write(pos);          // Le decimos al servo que vaya a la posición que
//tenemos declarada como "pos"
  servoMotor2.write(pos);
  delay(30);
}
for (pos = 120; pos >= 0; pos -= 2) { //Movimiento de 120 a 0°
  servoMotor1.write(pos);
  servoMotor2.write(pos);
  delay(30);
}

```

## **Camaleón**

```
/*
```

--> TCS3200 Sensor que reconoce el colores.

Los TCS3200 son sensores que convierten en frecuencia la intensidad de luz medida por una matriz de fotodiodos.

La frecuencia entregada por el sensor TCS3200 es mayor cuanto mayor luminosidad se detecte.

--> Los pines de conexión del sensor a Arduino se muestran en los comentarios

```

Sensor   Arduino
-----

```

```
VCC      5V
GND      GND
s0       8 // Entrada de selección de la escala de frecuencia de salida
s1       9 // Entrada de selección de la escala de frecuencia de salida
s2       10 // Entrada de selección del tipo de fotodiodo
s3       11 // Entrada de selección del tipo de fotodiodo
OUT      12
```

```
*/
```

```
//Pines del sensor que van conectado a arduino
```

```
const int s0 = 8;
const int s1 = 9;
const int s2 = 10;
const int s3 = 11;
const int out = 12;
```

```
/* La palabra clave const significa constante. Es un calificador variable que modifica el comportamiento de la variable,
```

```
haciendo que sea una variable de "sólo lectura". Esto significa que la variable se puede utilizar como cualquier otra
```

```
variable de este tipo, pero su valor no se puede cambiar. */
```

```
// Pines del led que van conectado a arduino
```

```
int redLed = 3; //izq
int greenLed = 6; //segundo de la derecha
int blueLed = 5; //primero de la derecha
```

```
// Variables
int red = 0;
int green = 0;
int blue = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);

  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(blueLed, OUTPUT);

  digitalWrite(s0, HIGH); // La escala de frecuencia de salida está S0 y S1 en high los
  dos, que sería al 100%.
  digitalWrite(s1, HIGH);
}

void loop()
{
  color();
  Serial.print(" Intensidad Rojo: ");
  Serial.print(red, DEC);
```

```
Serial.print(" Intensidad Verde: ");
Serial.print(green, DEC);
Serial.print(" Intensidad Azul: ");
Serial.print(blue, DEC);
//Serial.println();

// ----- Apagado ----- //
if (red == 47 || red == 49 || red == 50 || red == 42 || red == 48 && green == 47 ||
green == 52 || green == 53 || green == 54 && blue == 43 || blue == 44 || blue ==
42)
{
  Serial.println(" - (Apagado)");
  digitalWrite(redLed, LOW); // OFF
  digitalWrite(greenLed, LOW); // OFF
  digitalWrite(blueLed, LOW); // OFF
}

// ----- ROJO ----- //
else if (red == 11 || red == 12 && green == 35 || green == 36 || green == 37 ||
green == 30 && blue == 27 || blue == 26 || blue == 28)
{
  Serial.println(" - (Rojo)");
  digitalWrite(redLed, HIGH); // Led rojo ON
  digitalWrite(greenLed, LOW);
  digitalWrite(blueLed, LOW);
}

// ----- Verde ----- //
```

```
else if (red == 15 || red == 16 && green == 12 || green == 11 && blue == 16 ||
blue == 15)
{
  Serial.println(" - (Verde)");
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, HIGH); // Led verde ON
  digitalWrite(blueLed, LOW);
}

// ----- Azul ----- //

else if (red == 25 || red == 27 || red == 26 && green == 27 || green == 28 &&
blue == 18 || blue == 17 || blue == 12)
{
  Serial.println(" - (Azul)");
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, LOW);
  digitalWrite(blueLed, HIGH); // Led azul ON
}

/* // ----- Amarillo ----- //

else if (red == 5 || red == 6 && green == 9 && blue == 14 || blue == 15)
{
  Serial.println(" - (Amarillo)");
  digitalWrite(redLed, 255); // Led rojo ON
  digitalWrite(greenLed, 54); // Led verde ON
```

```
digitalWrite(blueLed, LOW);
}*/

// ----- Azul Cielo ----- //

else if (red == 19 || red == 20 && green == 11 || green == 12 && blue == 8)
{
  Serial.println(" - (Azul cielo)");
  digitalWrite(redLed, 0); // Led rojo OFF
  digitalWrite(greenLed, 191); // Led verde ON
  digitalWrite(blueLed, 255); // Led azul ON
}

// ----- Rosa ----- //

else if (red == 5 || red == 6 && green == 10 || green == 11 && blue == 7 || blue
== 8 || blue == 9)
{
  Serial.println(" - (ROSA)");
  digitalWrite(redLed, 255); // Led rojo ON
  digitalWrite(greenLed, 0);
  digitalWrite(blueLed, 128); // Led azul ON
}

else {
  Serial.println();
}

delay(300);
```

```
digitalWrite(redLed, LOW);
digitalWrite(greenLed, LOW);
digitalWrite(blueLed, LOW);
}
```

/\* Para seleccionar el color que se lee de los fotodiodos, se utilizan los pines de control S2 y S3.

Dado que los fotodiodos están conectados en paralelo, estableciendo S2 y S3 en diferentes combinaciones

de LOW y HIGH permite seleccionar diferentes grupos de fotodiodos. \*/

```
void color()
{
  digitalWrite(s2, LOW); // Al poner los dos en low seleccionamos el rojo
  digitalWrite(s3, LOW);
  //count OUT, pRed, RED
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); // Leyendo la
frecuencia de salida del sensor
  digitalWrite(s3, HIGH); // Al poner s3 en high se pone el azul
  //count OUT, pBLUE, BLUE
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); // Leyendo la
frecuencia de salida del sensor
  digitalWrite(s2, HIGH); // Al poner s2 en high se pone el verde
  //count OUT, pGreen, GREEN
  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); // Leyendo la
frecuencia de salida del sensor
}
```



## 8. Imágenes del Proyecto

- Lista de reproducción en Youtube de los videos del proyecto. Aquí se podrá ver el funcionamiento tanto de la maqueta del ojo como la del camaleón.

→ [https://www.youtube.com/playlist?list=PLgrxLfpCW8DUyISvUeHnaGFEfc1C\\_Q7Gg](https://www.youtube.com/playlist?list=PLgrxLfpCW8DUyISvUeHnaGFEfc1C_Q7Gg)

- Carpeta de drive con todas las fotos del proyecto.

→ <https://drive.google.com/open?id=1tEalYRg8yvyMUo1ZVnDh1h7a0BLknbjk>


Son muchas fotos y ponerlas aquí en el documento no lo veo adecuado, se ve pequeño y no se aprecia.

## 9. Incidencias

Respecto al ojo el principal problema que tuvimos fue encontrar un material para hacer el círculo porque no encontrábamos nada redondo que pudiésemos utilizar, y también un problema con un servo porque estaba roto y no lo sabíamos, al comprobar con otro material ya dimos con el fallo y sustituimos la pieza y todo funcionó a la perfección, por lo demás fue más o menos sencillo montarlo.

Respecto al camaleón tuvimos los siguientes problemas:

Empezamos con el diseño 3D del camaleón, buscamos el diseño por internet y lo tuvimos que modificar porque el camaleón venía completo en una misma pieza, nosotros lo necesitábamos en dos porque teníamos que meter los componentes dentro (leds) y después pegarlo para juntarlo. Lo hicimos con un programa de diseño 3D (Autodesk Meshmixer). Una vez tuvimos el diseño hecho contactamos con los compañeros de edificación para ver si nos podían imprimir en 3D la maqueta, y no podían porque la máquina estaba rota. En clase la máquina que teníamos también estaba rota. Llamamos a una empresa de Málaga (3Dzings) y nos cobraba casi 600€ por imprimir ambos diseños, la piedra y el camaleón, lo



descartamos totalmente. Contactamos con una antigua profesora y ella nos lo imprimió sin ningún problema. Además de la impresión de esta profesora, otro amigo de un compañero nos lo imprimió por 4€ y quedó bastante bien.

Una vez con el diseño del camaleón y de la piedra hechos nos dimos cuenta que la piedra nos quedaba muy pequeña, por lo que decidimos descartarla y comprar un baúl de madera. El baúl lo desmontamos, y le dimos con laca para madera, en un principio, para oscurecer la caja y que tomara un efecto de tronco de árbol. Después nos dimos cuenta que la caja no se iba a oscurecer porque era laca transparente por lo que tuvimos que lijar toda la caja para posteriormente aplicarle una especie de colorante natural que nos dejó prestado el profesor de arte. Una vez aplicado la caja quedó bastante bien.

Ya teníamos el diseño del camaleón (que por cierto, una pata se rompió y tuvimos que pegarla) y la caja terminada, ahora nos quedaba empezar con la programación y las pruebas.

Estuvimos probando con tiras leds, pero al final nos decantamos por leds rgb que eran más sencillo de conectar y la programación se hacía más fácil.

El sensor de color tcs3200 nos llegó 2 días antes de entregar el proyecto, porque lo pedimos por nuestra cuenta ya que el material no nos llegaba así que íbamos con el tiempo muy justo para entregarlo a tiempo. Lo montamos, y después de probar muchísimos códigos dimos con uno que nos daba los 3 colores principales RGB. Lo tuvimos que calibrar para que nos diese los demás colores. De tanto probar se rompieron varios leds RGB.

Íbamos a usarlo a batería pero nos dimos cuenta que a la hora de probar el calibrado variaba y no conseguimos que se iluminase de los colores que detectaba así que decidimos dejarlo conectado al cable de arduino que funcionaba perfecto.

En resumen, el proyecto ha sido toda una aventura, a pesar de no tener un funcionamiento muy complejo a nosotros todo el proceso nos ha parecido muy divertido.

## 10. Web utilizadas

- <https://www.instagram.com/p/B2tWIPFoFxS/> ← Idea camaleón saca de aquí
- <https://www.instagram.com/p/B7EceZ8olfm/> ← Funcionamiento aquí
- <https://www.luisllamas.es/arduino-led-rgb-ws2812b/>
- <https://www.youtube.com/watch?v=940Nole57Mk>
- <https://www.youtube.com/watch?v=Tm-lZavMvaQ>
- <http://blascarr.com/lessons/tcs3200-color-sensor/>
- <https://www.luisllamas.es/medir-color-arduino-colorimetro-tcs3200/>
- <https://www.prometec.net/rgb-led/>
- <http://robots-argentina.com.ar/didactica/arduino-reconocer-colores-con-el-modulo-tcs230>
- [www.electronicshobby.com/electronic-projects/rgb-color-detector-tcs3200-sensor-module](http://www.electronicshobby.com/electronic-projects/rgb-color-detector-tcs3200-sensor-module)
- <https://create.arduino.cc/projecthub/vishwasnavada/electronic-chameleon-d9acd9>
- [https://www.prometec.net/wp-content/uploads/2014/10/Sesion11\\_RB\\_LED\\_bb.jpg](https://www.prometec.net/wp-content/uploads/2014/10/Sesion11_RB_LED_bb.jpg)
- <https://surtrtech.files.wordpress.com/2019/08/wiring-2.png>
- <https://miarduinounotieneunblog.blogspot.com/2016/02/camaleon-con-sensor-de-reconocimiento.html>
- <https://cults3d.com/es/modelo-3d/juegos/3d-scanned-kitzbuhel-rock> - diseño piedra impresa en 3d que al final no usamos
- <https://www.thingiverse.com/thing:628911> - Diseño impresión 3d del camaleón usada.