

# HEXÁGONO CINÉTICO

Romero Vílchez, Celia María  
S21ME-MEEI  
13/02/2022

## **ÍNDICE**

<b>Descripción</b>	<b>2</b>
<b>Electrónica</b>	<b>2</b>
<b>Montaje</b>	<b>3</b>
<b>Programación</b>	<b>5</b>
<b>Incidencias</b>	<b>7</b>
<b>Materiales</b>	<b>8</b>
<b>Enlaces de interés</b>	<b>8</b>

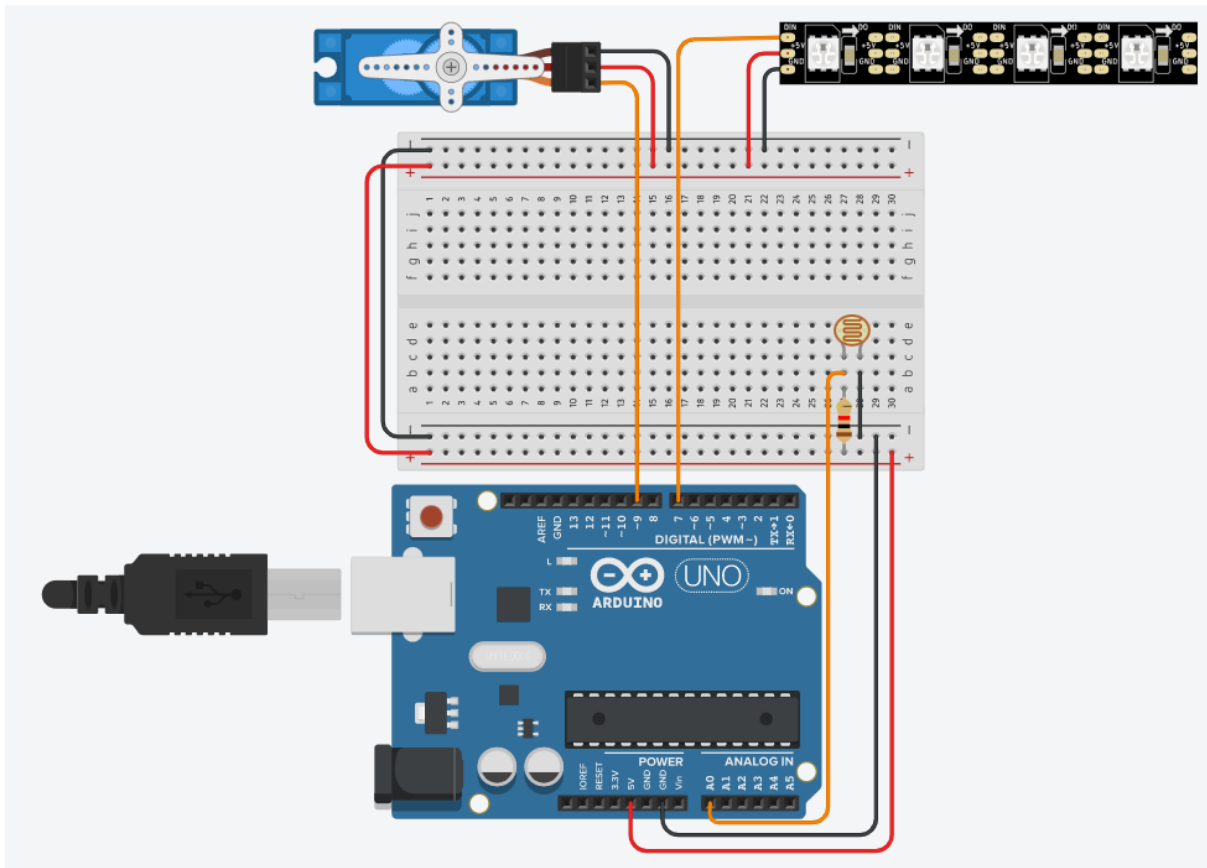
## 1. Descripción

En colaboración con Bachillerato de Artes, se ha realizado una escultura con forma de ventana. Su propósito es representar cómo influyen hoy día las redes sociales en nuestra vida.

El proyecto trata de una estructura con forma hexagonal, simulando la forma y el mecanismo del obturador de una cámara fotográfica. Cuando el obturador está cerrado se representa una imagen. Mediante una fotorresistencia LDR se activa el mecanismo cuando se detecta una luz, el flash de un móvil por ejemplo. De esta forma el obturador queda abierto mostrando una segunda imagen, la cual representa la realidad detrás de lo que se nos muestra a través de las redes.

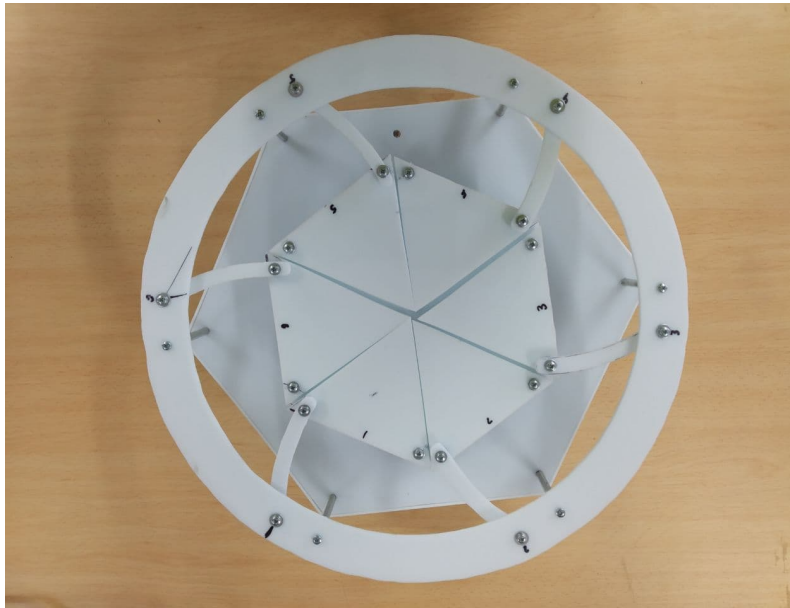
El mecanismo va acompañado de luces LEDs decorativas que pueden ser reprogramadas en distintos colores en función del arte.

## 2. Electrónica

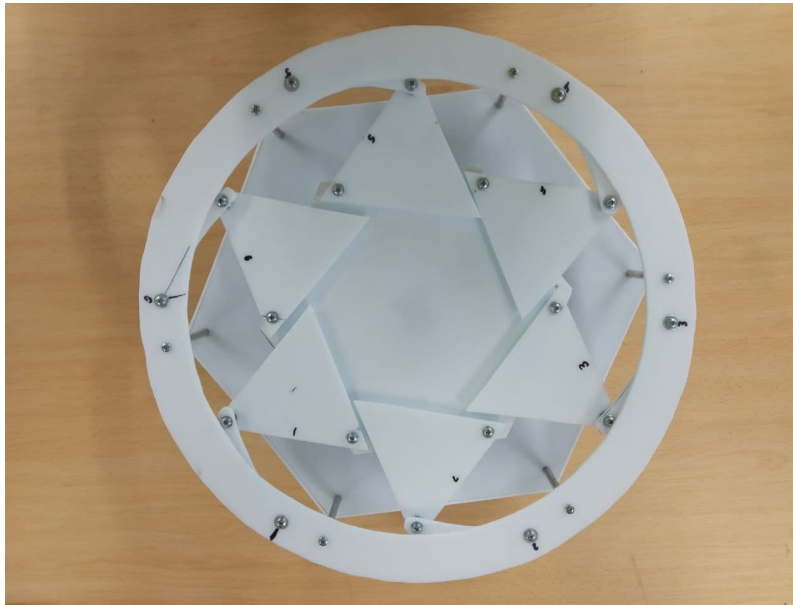


### 3. Montaje

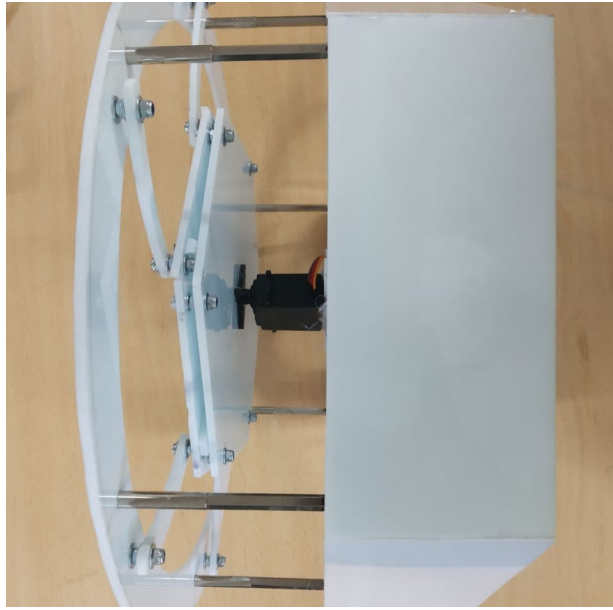
Mecanismo cerrado:



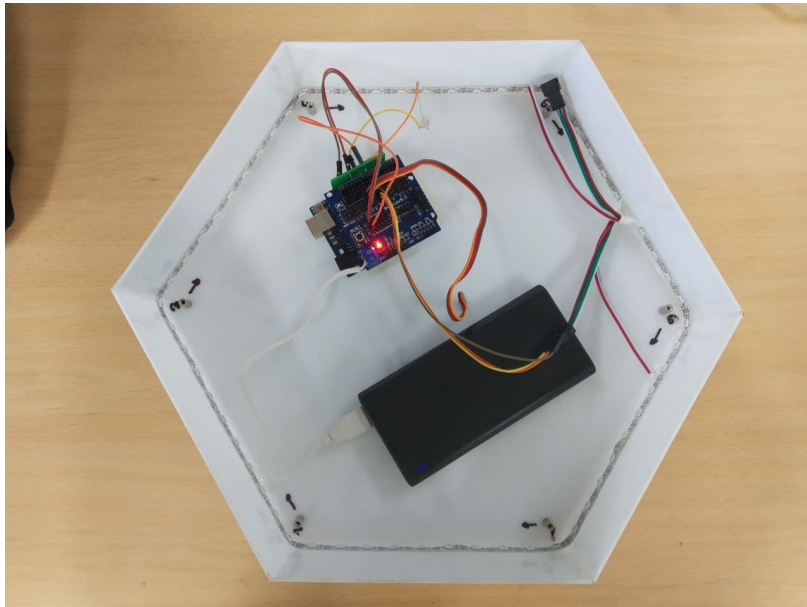
Mecanismo abierto:



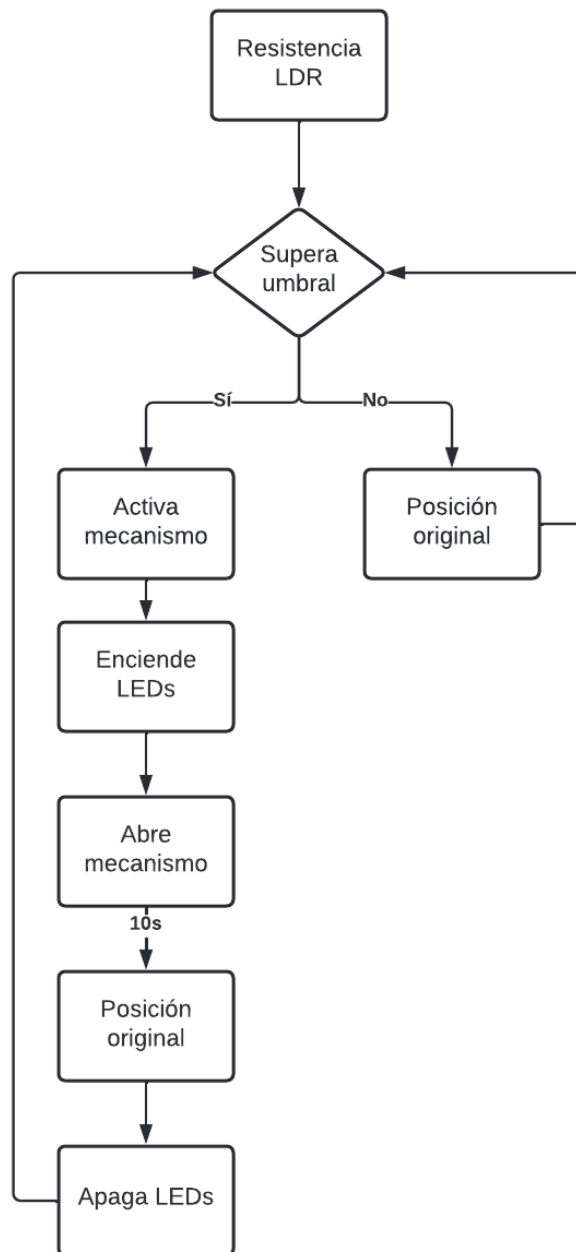
Mecanismo vista lateral:



Vista inferior electrónica:



## 4. Programación



```
#include <Servo.h>
#include <Adafruit_NeoPixel.h>

#define PINLEDRGB 7 //pin data 12
#define NUMPIXELS 60 //numero de leds en la tira

int LDR = A0;
int valorLDR = 0;
```

```

Servo myservo; // create servo object to control a servo
Adafruit_NeoPixel pixels(NUMPIXELS, PINLEDRGB, NEO_GRB + NEO_KHZ800);

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
  myservo.write(30);
  pixels.begin(); //Inicializamos el objeto "pixels"
}

void loop() {
  valorLDR = analogRead(LDR);
  Serial.println(valorLDR);

  if (valorLDR < 500) {
    pixels.clear(); //limpiamos el color

    for (int i = 0; i < NUMPIXELS; i++) {
      pixels.setPixelColor(i, pixels.Color(0, 150, 0)); // Brillo
moderado en rojo
      pixels.show(); // Mostramos y actualizamos el color del pixel de
nuestra cinta led RGB
      delay(50); // Pausa por un periodo de tiempo (en milisegundos).
    }
    delay(2000);

    for (int i = 0; i < NUMPIXELS; i++) {

      pixels.setPixelColor(i, pixels.Color(150, 0, 0)); // Brillo
moderado en rojo

      pixels.show(); // Mostramos y actualizamos el color del pixel de
nuestra cinta led RGB

      delay(50); // Pausa por un periodo de tiempo (en milisegundos).
    }
    for (int pos = 30; pos >= 0; pos -= 5) { // goes from 30 degrees to
0 degrees
      myservo.write(pos); // tell servo to go to position
in variable 'pos'
      delay(15); // waits 15 ms for the servo to
reach the position
    }
    delay (10000);
    for (int pos = 0; pos <= 30; pos += 5) { // goes from 0 degrees to

```

```

30 degrees
    // in steps of 1 degree
    myservo.write(pos);           // tell servo to go to position
in variable 'pos'
    delay(15);                    // waits 15 ms for the servo to
reach the position
}

for (int i = 0; i < NUMPIXELS; i++) {

    pixels.setPixelColor(i, pixels.Color(0, 150, 0)); // Brillo
moderado en rojo

    pixels.show(); // Mostramos y actualizamos el color del pixel de
nuestra cinta led RGB

    delay(50); // Pausa por un periodo de tiempo (en milisegundos).

}

pixels.clear();
for (int i = 0; i < NUMPIXELS; i++) {

    pixels.setPixelColor(i, pixels.Color(0, 0, 0)); // Brillo moderado
en rojo

    pixels.show(); // Mostramos y actualizamos el color del pixel de
nuestra cinta led RGB

    delay(50); // Pausa por un periodo de tiempo (en milisegundos).
}

```

## 5. Incidencias

El diseño final del proyecto ha ido cambiando a lo largo del desarrollo del mismo.

La idea original tenía un mecanismo puesto en marcha mediante 3 servomotores, que con el progreso de su montaje no supe resolver, por lo que decidí cambiar el diseño de forma que pudiera usar los mismos materiales que en el original.

A la hora del montaje tuve que volver a recortar tanto los triángulos que conforman el obturador como las varillas que mueven el mecanismo ya que las mediciones no eran las correctas, y esto creaba conflicto en el movimiento.



La caja para el mecanismo se pretendía situar encima del hexágono base, pero a la hora de montarlo, los brazos chocaban con las paredes por lo que se ha procedido a situarla debajo del hexágono base.

Existe un conflicto de programación que proporciona corriente al servomotor antes y después de su activación. No interfiere en su funcionamiento final.

## 6. Materiales

UD	NOMBRE	DESCRIPCIÓN	QTY	PRECIO UD	PRECIO TOTAL
1	Arduino	Arduino Uno	1	11,59	11,59
2	Servomotor	MG996R	1	2,59€	2,59€
3	Resistencia	4k7Ω	1	0,10€	0,10€
4	Resistencia	LDR	1	0,15€	0,15€
5	LED	WS2812B 1m 60leds	1	5,49€	5,49€
6	Lámina Propileno	3.5mm 100x100cm	1	17,49€	17,49€
7	Tornillería	Tornillo 3mm pack x30	1	1,99€	1,99€
8	Tornillería	Arandela 3mm pack x30	1	1,99€	1,99€
9	Tornillería	Tuerca 3mm pack x30	1	1,99€	1,99€
10	Tornillería	Tuerca de seguridad 3mm pack x30	1	1,99€	1,99€
11	Tornillería	Separador hexagonal	6	0,54€	3,24€
<b>Total</b>					48,61

## 7. Enlaces de interés

[https://www.youtube.com/watch?v=jeTtE9\\_y8Ow&t=45s](https://www.youtube.com/watch?v=jeTtE9_y8Ow&t=45s)

<https://www.youtube.com/watch?v=zQ9FAwAmt98>

<https://www.imarabe.org/en/architecture>