

# **Informe Parcial 2**

Informática II

**Daniel Perez Gallego CC. 1193088770**

**Jorge Montaña Cisneros CC.**

**1007327968**

Departamento de Ingeniería Electrónica y

Telecomunicaciones

Universidad de Antioquia

Medellín

Septiembre de 2021

# Índice

<b>1. Clases implementadas</b>	<b>1</b>
1.1. Menu . . . . .	1
1.2. Imagen . . . . .	1
1.3. Pixel RGB . . . . .	1
<b>2. Esquema de las clases</b>	<b>2</b>
<b>3. Módulos de código de interacción</b>	<b>2</b>
<b>4. Estructura del circuito montado</b>	<b>4</b>
<b>5. Problemas presentados</b>	<b>4</b>

## 1. Clases implementadas

### 1.1. Menu

Clase interactiva con el usuario, delegada de pedir el nombre de la imagen con su formato, almacenada en la carpeta 'Imágenes'. Retorna la variable 'im', correspondiente a la imagen cargada con el tipo QImage.

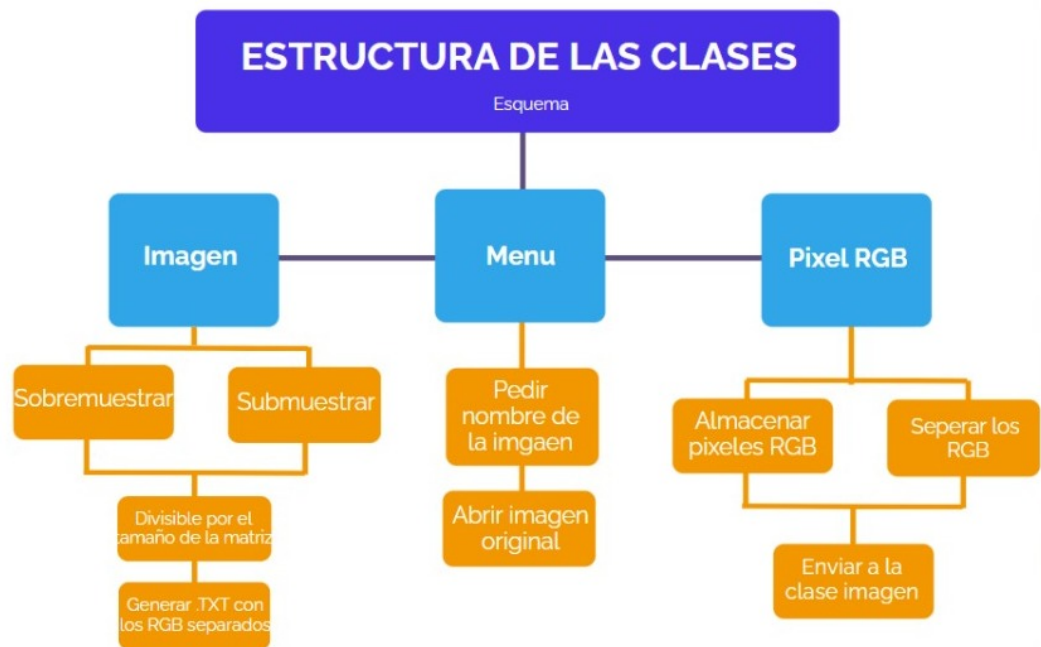
### 1.2. Imagen

Encargada de manejar la imagen con los parámetros de su alto y ancho, promediar los colores por bloques y crear el .txt generado con el formato adecuado.

### 1.3. Pixel RGB

Es la encargada de almacenar los pixeles RGB de la imagen para luego separarlos, tiene como parámetro los colores del RGB

## 2. Esquema de las clases



## 3. Módulos de código de interacción

```
1 class Imagen
2 {
3 private:
4     int fila, columna;
5     vector<vector<Pixel_RGB>> Pixel_color;
6 public:
7     Imagen();
8     Imagen(int M, int N);
9     void set_color(int x, int y, Pixel_RGB color);
10    int getFila() const;
11    void setFila(int value);
12    int getColumna() const;
13    void setColumna(int value);
14    void imprimir_pruebas();
15    void txt_generado();
16    Pixel_RGB Promedio_Color(int fo, int cantidadF, int co, int
    cantidadC);
17    Pixel_RGB recorrer(int fo, int co);
18 };
```

Listing 1: Parámetros de la clase imagen

```

1 Pixel_RGB Imagen::recorrer (int fo, int co)
2 {
3     int limF = fo;
4     int limC = co;
5     int Red = 0, Green= 0, Blue = 0;
6     for (int f=fo; f<=limF; f++ ) {
7         for (int c=co; c<=limC; c++ ) {
8             Red = Pixel_color[f][c].getRed();
9             Green = Pixel_color[f][c].getGreen();
10            Blue = Pixel_color[f][c].getBlue();
11        }
12    }
13
14    return Pixel_RGB(Red, Green, Blue);
15 }

```

Listing 2: Clase recorrer

```

1 Pixel_RGB Imagen::Promedio_Color(int fo, int cantidadF, int co, int
  cantidadC)
2 {
3     int limF = fo+cantidadF;
4     int limC = co+cantidadC;
5     int pixeles = cantidadF*cantidadC;
6     int sumaRed = 0, sumaGreen= 0, sumaBlue = 0;
7     for (int f=fo; f<limF; f++ ) {
8         for (int c=co; c<limC; c++ ) {
9             sumaRed += Pixel_color[f][c].getRed();
10            sumaGreen += Pixel_color[f][c].getGreen();
11            sumaBlue += Pixel_color[f][c].getBlue();
12        }
13    }
14
15    return Pixel_RGB(sumaRed/pixeles, sumaGreen/pixeles, sumaBlue/
  pixeles);
16 }

```

Listing 3: Clase promedio color

```

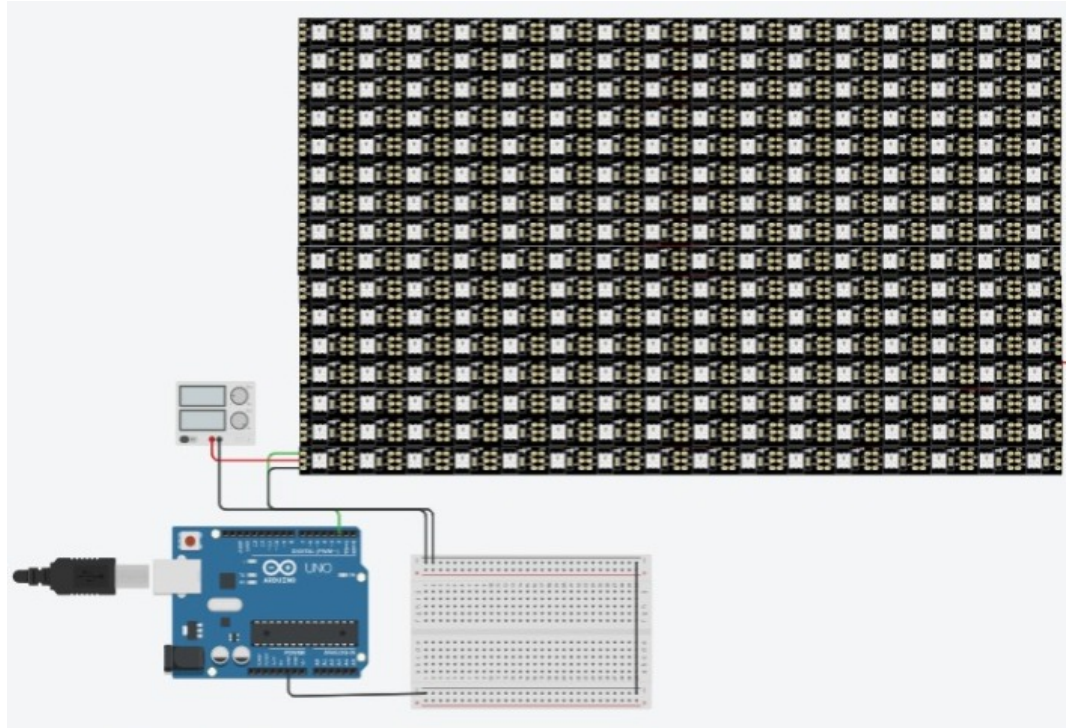
1 void Imagen::set_color(int x, int y, Pixel_RGB color){
2     Pixel_color[x][y].setColor(color);
3 }

```

Listing 4: Clase set color

## 4. Estructura del circuito montado

Para la matriz de LEDs en Tinkercad, diseñamos un circuito de 16x16 LEDs, hecha con tiras de neopixel. Cada una con su salida conectada a la entrada de la fila/tira superior, la potencia conectada a un suministro de energía y todas las conexiones para que el circuito funcione con normalidad



## 5. Problemas presentados

Justo como lo analizamos, el método para reducir y amplificar la imagen fué la parte más complicada en la implementación, a pesar de que buscamos varias métodos, a la hora de codificarlo se complicaba y comenzamos a buscar un método para simplificarlo, hasta el punto donde consideramos aplicar un nuevo método y empezar casi desde 0.

Problemas para la función de sobremuestreo, con los bloques impares.

Desconocíamos el formato que debían ser escritos los RGB en el .txt generado para tinkercad y si teníamos que insertar algún método para que el usuario no tenga que copiar y pegar el RGB en el tinkercad

La conexión del circuito fué un problema menor gracias a la búsqueda de documentación y videos sobre el código y la conexión en tinkercad; sin embargo, pensábamos que se encenderían los LEDs rápido, pero como no lo hacian debido a toda la información que se procesaba, abortabamos el proceso pensando que el circuito estaba malo, pero no lo estaba, solo éramos muy impacientes.