

# **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

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Análisis</b>                        | <b>1</b> |
| 1.1      | Análisis del problema . . . . .        | 1        |
| 1.2      | Tareas a realizar . . . . .            | 2        |
| 1.3      | Algoritmo implementado . . . . .       | 2        |
| 1.4      | Consideraciones . . . . .              | 3        |
| <b>2</b> | <b>Clases implementadas</b>            | <b>4</b> |
| 2.1      | Imagen . . . . .                       | 4        |
| 2.2      | Pixel RGB . . . . .                    | 4        |
| <b>3</b> | <b>Esquema de las clases</b>           | <b>4</b> |
| <b>4</b> | <b>Código</b>                          | <b>4</b> |
| <b>5</b> | <b>Estructura del circuito montado</b> | <b>4</b> |
| <b>6</b> | <b>Problemas presentados</b>           | <b>5</b> |

## 1 Análisis

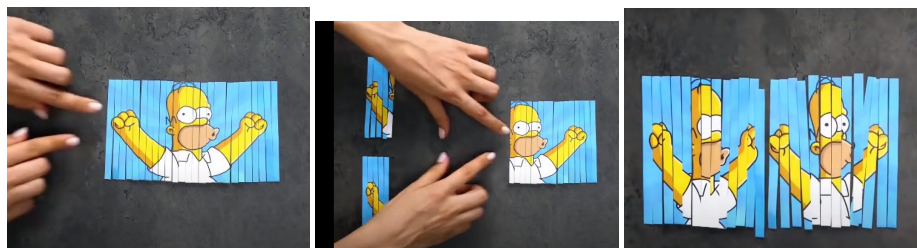
### 1.1 Análisis del problema

Al analizar detenidamente el parcial y las instrucciones planteadas, observamos que el mayor reto consistiría en la modificación del tamaño de las imágenes, adaptándolo a un tamaño específico, ya sea aumentando o disminuyendo la proporción.

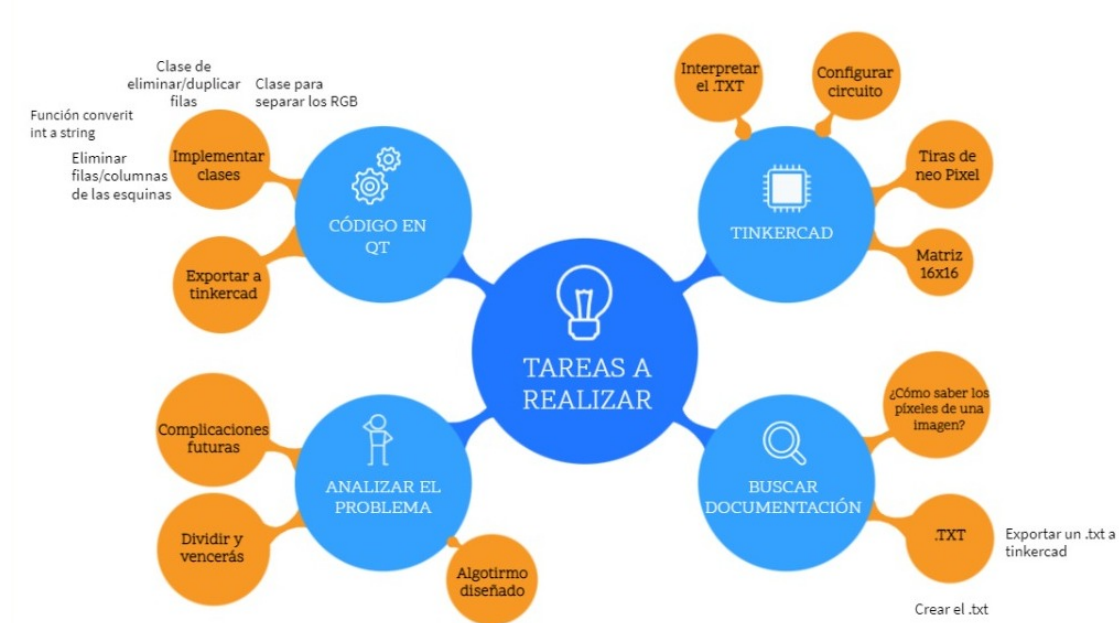
Si la imagen dada tiene un tamaño menor al esperado, se hará un proceso parecido al de eliminar, pero en este caso se aumentarán las filas/columnas colocandolas continuas a ellas mismas hasta quedar con la proporción deseada.

Para reducir el tamaño de la imagen, implementamo un método para dividir el número de filas y columnas de la imagen original por el tamaño deseado.

Mientras más pequeña sea la matriz de LEDs, menos información deberemos exportar, será más eficiente y fácil, sin embargo, la imagen se volverá difícil de reconocer para el usuario.

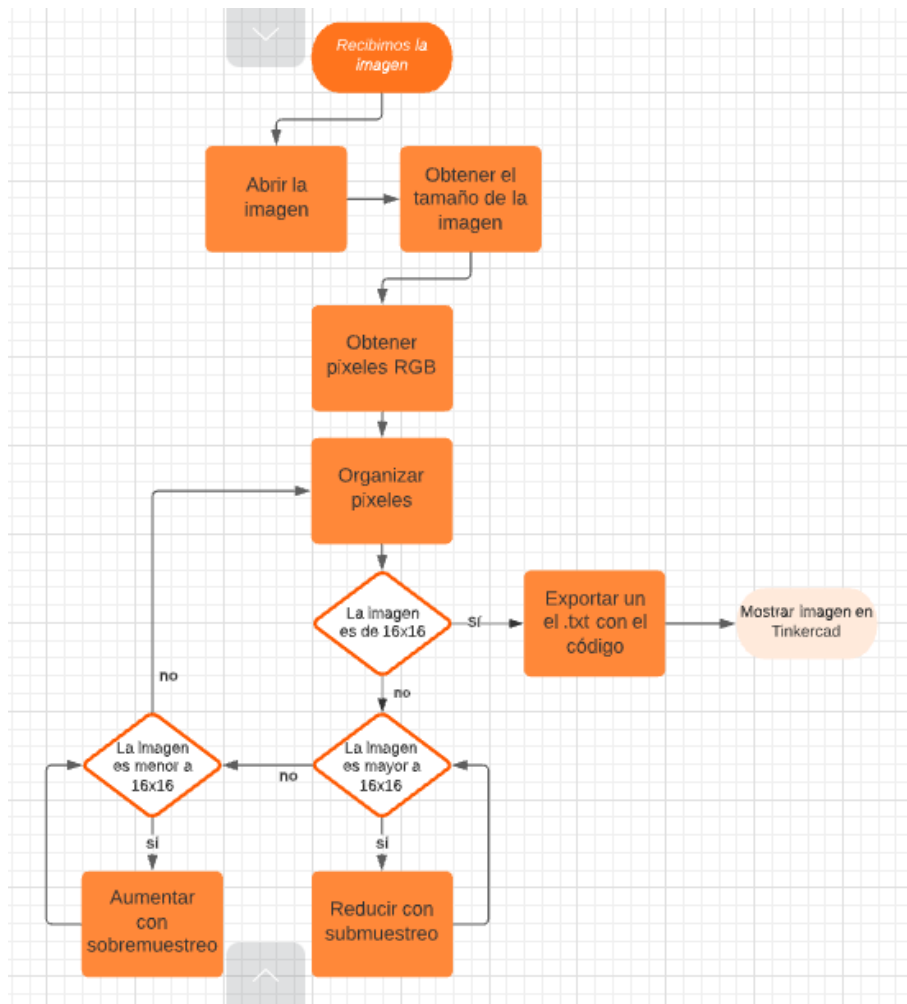


## 1.2 Tareas a realizar



## 1.3 Algoritmo implementado

Diagrama de flujo del algoritmo que implementaremos para la solución del problema (No código)



## 1.4 Consideraciones

Una de las consideraciones más importantes que encontramos fué una correcta identificación de cada columna y fila de píxeles RGB, y mirar el modo de separar cada una.

El método para generar el .txt final y enviarlo a tinkercad.

## 2 Clases implementadas

### 2.1 Imagen

Encargada de manejar la imagen

### 2.2 Pixel RGB

Es la encargada de almacenar los pixeles RGB de la imagen

## 3 Esquema de las clases

## 4 Código

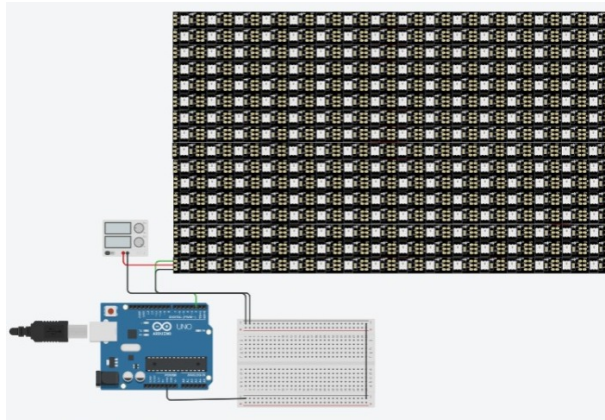
```
string filename="../ParcialInfo2/imagenes/Brasil.png";
QImage im(filename.c_str());

vector<vector<vector<int>>> arreglo3d(im.width(), vector<vector<int>>(im.height(), vector<int>(3)));

for(int filas=0; filas<im.width(); ++filas){
    for(int columnas=0; columnas<im.height(); ++columnas){
        for(int m=0; m<3; m++){
            if(m==0){
                arreglo3d[filas][columnas][m]=im.pixelColor(filas,columnas).red();
            }
            else if(m==1){
                arreglo3d[filas][columnas][m]=im.pixelColor(filas,columnas).green();
            }
            else if(m==2){
                arreglo3d[filas][columnas][m]=im.pixelColor(filas,columnas).blue();
            }
        }
    }
}
```

## 5 Estructura del circuito montado

Para la matriz de LEDs en Tinkerdad, 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



## 6 Problemas presentados