

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

1	Análisis	1
1.1	Análisis del problema	1
1.2	Tareas a realizar	2
1.3	Algoritmo implementado	2
1.4	Consideraciones	3
2	Clases implementadas	4
3	Esquema de las clases	4
4	Código	4
5	Estructura del circuito montado	4
6	Problemas presentados	4

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.

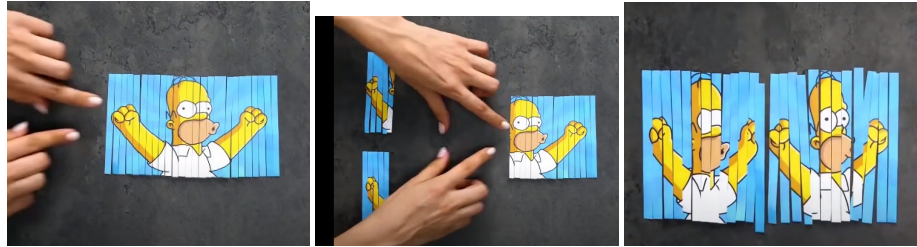
Para el submuestreo, pensamos a forma de solución organizar los RGB en filas y columnas separadas, una vez hecho esto, eliminar las que son pares, de este modo, tendremos la misma imagen, pero recortada a un cuarto de la, obteniendo un tamaño menor pero proporcional a la imagen original, repiendo el proceso hasta obtener el tamaño deseado.

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.

En el caso de que la imagen no se adapte al tamaño específico deseado, se va a eliminar/aumentar ya sea las filas o columnas sobrantes o faltantes de los extremos para que tenga una proporción adecuada.

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, por lo tanto, acordamos hacer la matriz de LEDs de 16x16

Para las clases, usaremos una que nos entregue los píxeles RGB y lo entregue en contenedores. Otra clase para eliminar/duplicar filas y columnas además, una clase para eliminar/agregar las filas/columnas en los extremos para llegar al tamaño de la matriz deseado. Y para finalizar, otra clase para convertir las variables a string y con esto último, generar un .txt

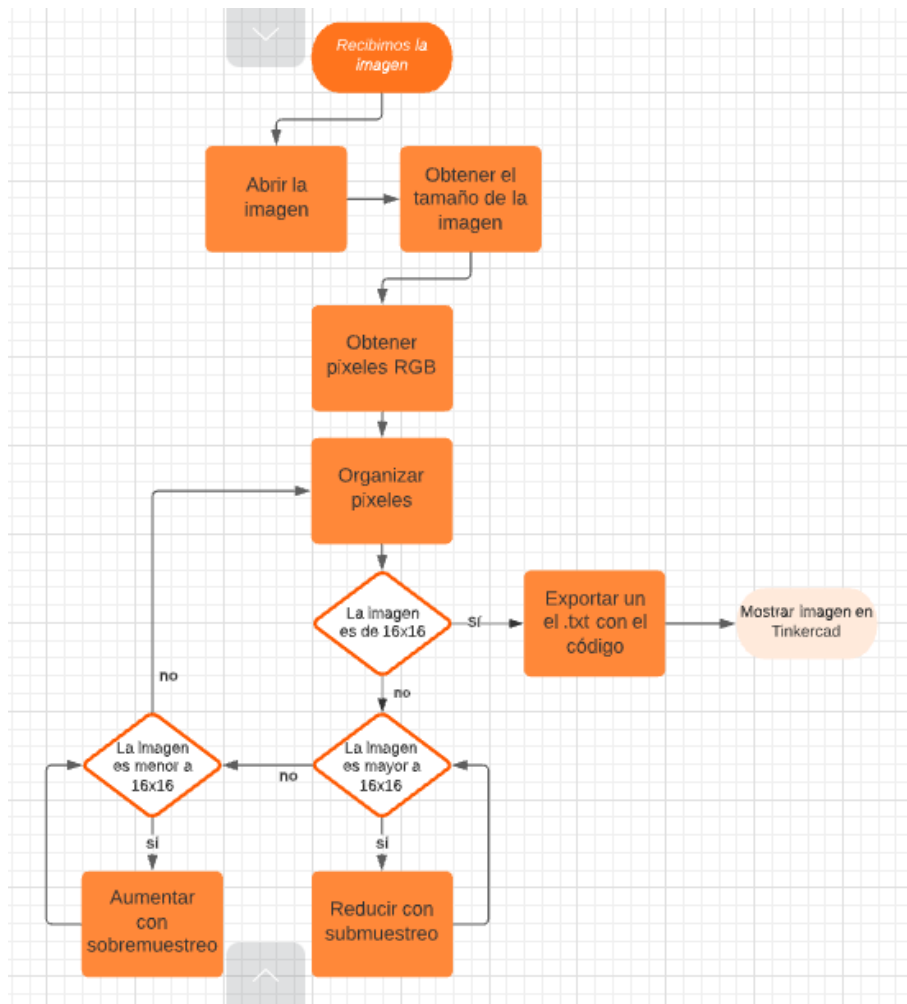


1.2 Tareas a realizar



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

3 Esquema de las clases

4 Código

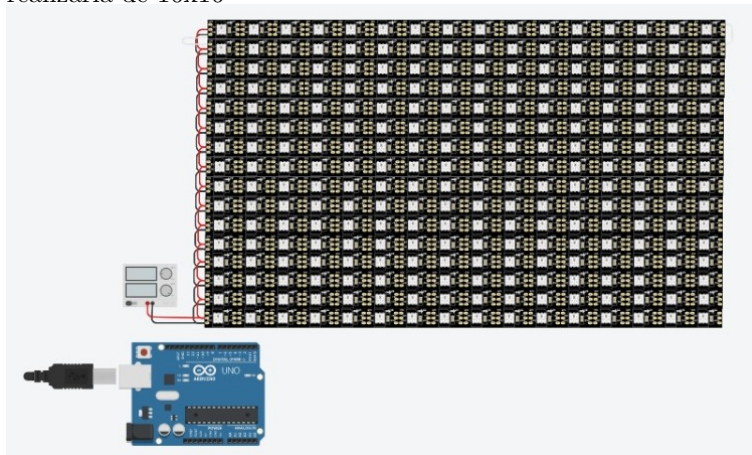
```
string filename="../ParcialInfo2/imagenes/Brasil.png";
QImage im(filename.c_str());

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

Nuestro primer diseño de la matriz de LEDs en Tinkercad, el cual pensamos realizarla de 16x16



6 Problemas presentados