

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

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
2.1. Imagen	4
2.2. Pixel RGB	4
3. Esquema de las clases	4
4. Código	4
5. Estructura del circuito montado	4
6. Problemas presentados	5

1. Análisis

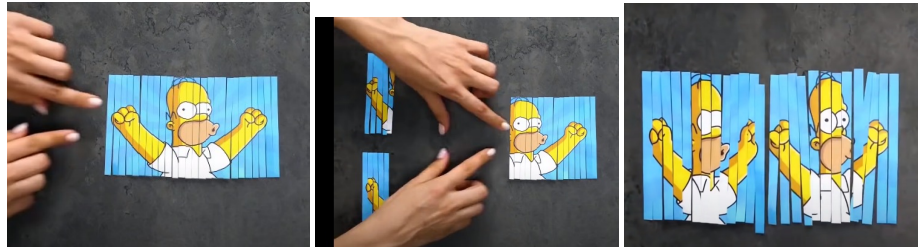
1.1. Análisis del problema

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.

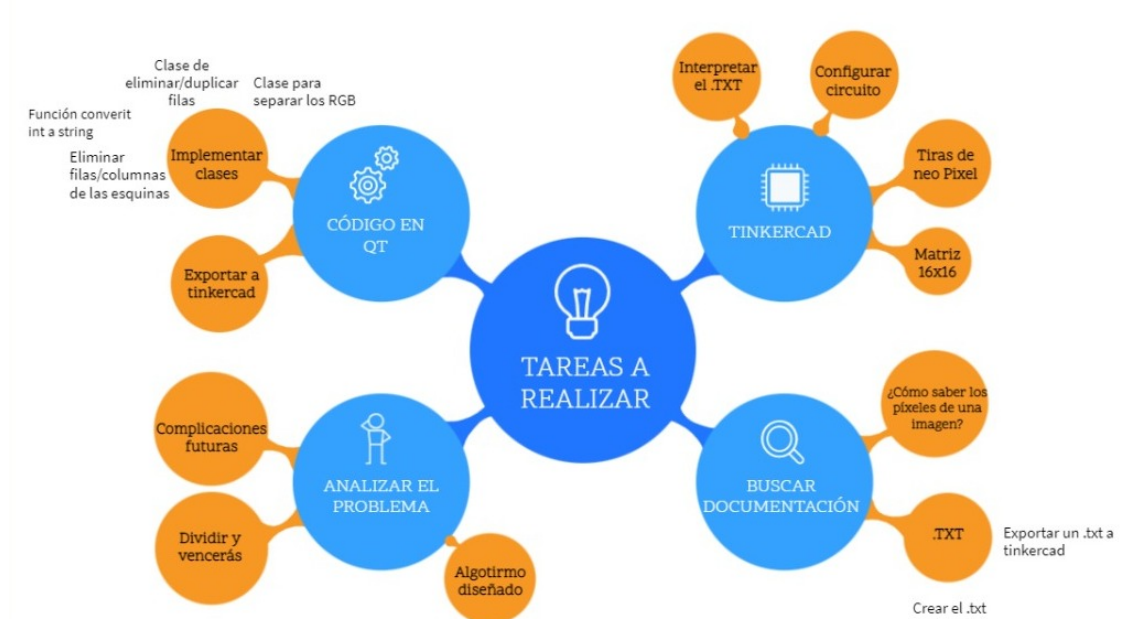
Si la imagen dada tiene un tamaño menor al esperado, se hará un proceso parecido al de eliminar, pero en este caso se aumentarán las filas/columnas colocandolas continuas a ellas mismas hasta quedar con la proporción deseada.

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.

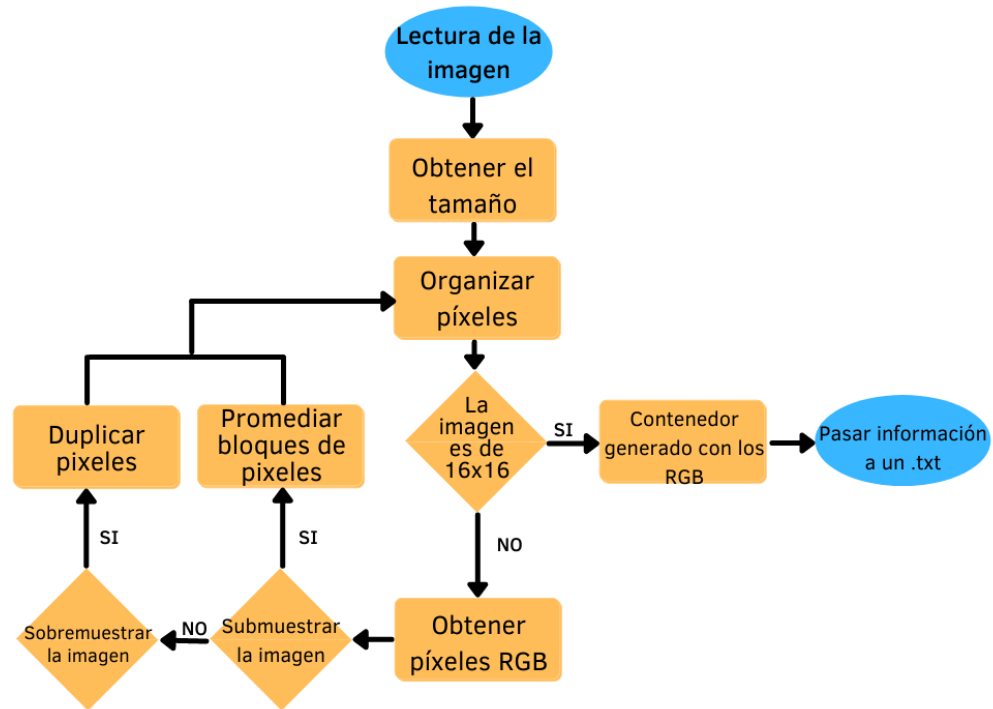


1.2. Tareas a realizar



1.3. Algoritmo implementado

Diagrama de flujo del algoritmo que implemenatremos 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 y el formato para generar el .txt final y enviarlo a tinkercad.

Si una imagen tiene tamaños que necesitan procesos distintos, como una que sea 10x100, se tendrán que realizar los dos procesos de sobremuestreo y submuestreo

Si se puede generar un inconveniente si se pone una imagen con un formato distinto a jpg como png o jpeg

2. Clases implementadas

2.1. Imagen

Encargada de manejar la imagen con los parámetros de su alto y ancho

2.2. Pixel RGB

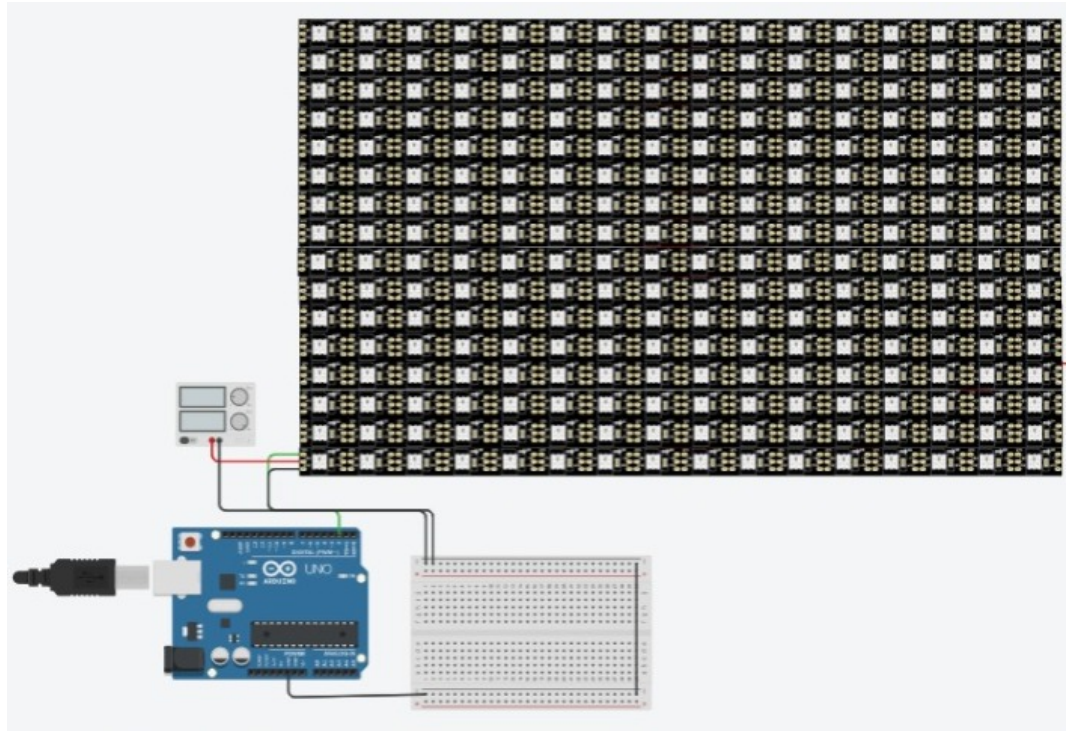
Es la encargada de almacenar los pixeles RGB de la imagen para luego separarlos

3. Esquema de las clases

4. Código

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

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.

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, sin embargo, acudimos a ver más documentación y videos sobre el código y la conexión de Tinkercad más de lo que pensábamos.