

Informa2 S.A.S

Parcial 2

Ferney Mejía Pérez
Marcela Flórez Orellano

**Implementación de la solución planteada
para el desafío**

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
28 septiembre de 2021

Índice

1. Sección de contenido	2
1.1. Solución del problema	2
1.2. Clases implementadas	2
1.3. Esquema de la estructura final de las clases	3
1.4. Tareas	3
1.5. Módulos de código implementados	4
1.6. Circuito	4
1.7. Manual de uso de nuestro programa	5
1.8. Problemas presentados	5

1. Sección de contenido

1.1. Solución del problema

Para la solución del problema se hizo control sobre una matriz 12x12 a partir de tiras de neopixels.

Primero que todo, se hizo una lectura minuciosa sobre los diferentes puntos o requisitos a considerar para la entrega de este trabajo, el cual consiste en presentar en una pantalla con leds RGB la nacionalidad de los competidores que han llegado al podio de triunfadores.

Para la solución del problema pensamos en emplear listas con el objetivo de almacenar los datos proporcionados por la función color pixel y a su vez ser capaces de realizar varias modificaciones sobre el contenedor.

Para realizar la compresión de la imagen usando la función submuestrear se va a realizar un promedio entre los valores contiguos de la matriz con el fin de eliminar ya sea las columnas o las filas.

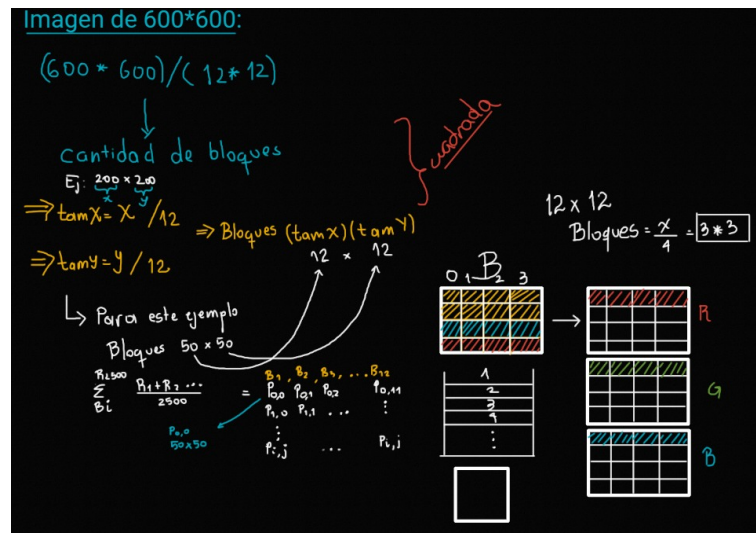
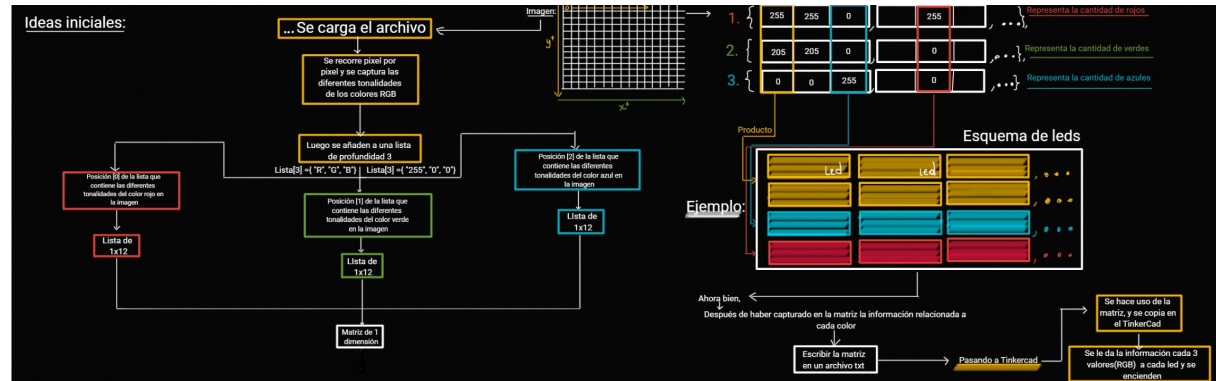
Para ampliar la imagen usando la función sobremuestrear se va a realizar una copia de los valores de la matriz y así, agregar elementos en las columnas y las filas.

La decisión del cambio de estructura de mapa a lista y vector, se debe a qué la accesibilidad en los mapas se debe de dar con clave y valor, dónde inicialmente se tenía clave RGB para cada color, pero nos percatamos que la accesibilidad mediante listas era más práctico y se optimizaba la ejecución de lectura y creación de datos finales en el Qt, todo pensando para llevarlo al Tinkercad.

1.2. Clases implementadas

Para el funcionamiento del programa se creó una clase de nombre RGB, la cual se encarga de todo lo relacionado con el procesamiento de la imagen suministrada.

1.3. Esquema de la estructura final de las clases



1.4. Tareas

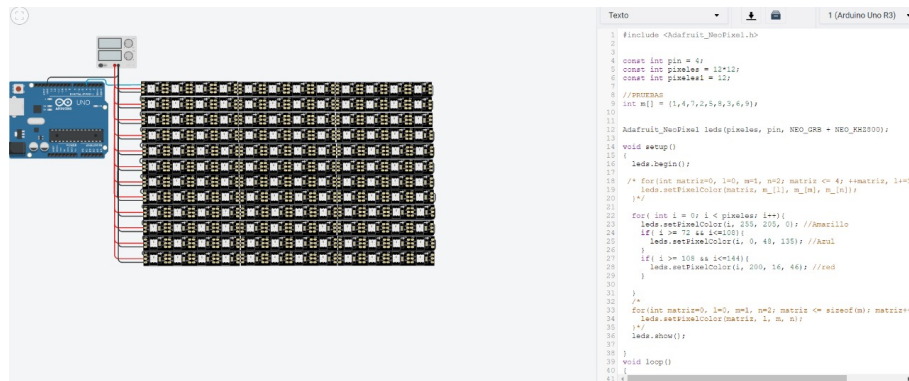
- Cargar la imagen.
- Desarrollar función de proceso de colores en pro a la creación de una lista de lista con profundidad 3.
- Desarrollar función que lee la matriz y llevarla a un vector.
- Desarrollo de la función de sub muestreo.

1.5. Módulos de código implementados

La clase RGB presenta las siguientes funciones públicas:

- CapturaRGB, la cual recibe como parámetro un QImage imagen”. A partir de un for anidado se recorre cada pixel de la imagen con el objetivo de extraer las diferentes intensidades de los componentes RGB y almacenar cada uno en un entero para ser luego llevado a una lista.
- Lectura matrix, recibe como parámetro una lista de listas, en la cual su lista interna tiene una profundidad tres. Se implementaron dos iteradores, uno que apunta al cascarón y otro que apunta a ese que se mueve de dimensión tres, con el fin de sacar la información contenida en cada uno y se retorna un vector con los valores obtenidos.
- SubMostreo, recibe como parámetros las dimensiones de la imagen y una lista que contiene una lista de profundidad tres, tiene como finalidad retornar una lista. En esta función se definen varios ciclos for, que dependen de un ciclo while, cada for arroja el promedio de cada color R,G,B, además, los acumuladores reciben la indexación de la posición del vector para poder sacar el promedio, esto reiterativamente hasta ir incrementando los valores del ciclo while.
- EscribirArchivo, recibe como parámetros el nombre del archivo .txt donde se va a guardar la información y una lista con los datos de las intensidades, la función se encarga de la escritura de la información obtenida en un archivo.

1.6. Circuito



1.7. Manual de uso de nuestro programa

- Primero que todo el programa le pedirá al usuario la ubicación de la imagen de la bandera.
- Se realizará el procesamiento de la información de la imagen, a partir de los datos suministrados por cada pixel de la misma.
- De ser necesario se efectuará un ajuste de sus dimensiones para ser presentada en la matriz 12x12 de neo píxeles creada en Tinkercad.
- Se usará la función de sub muestreo en el caso de que las dimensiones de la imagen sean mayores a las dimensiones de nuestra matriz 12x12 y la función sobre muestreo en caso de que la imagen tenga dimensiones menores.
- Al finalizar el proceso, el programa deberá generar un archivo de formato .txt con los datos modificados de cada intensidad de los componentes RGB, y así poder ser agregado al código de Arduino.
- La información del archivo .txt se copia y se pega en una matriz de enteros en Tinkercad.
- A partir de un ciclo for se encienden los leds de la matriz en Tinkercad.

1.8. Problemas presentados

Durante el desarrollo del parcial se presentaron problemas relacionados con la estructura de la solución del mismo, se pensaron e intentaron diferentes formas sin embargo no se llegó al resultado deseado.