

INFORMA2 S.A.S

Parcial 2: Análisis y diseño

Juan Pablo Cruz Gómez

Erika Dayana León Quiroga

Departamento de Ingeniería Electrónica y Telecomunicaciones
Universidad de Antioquia
Medellín
Septiembre de 2021

Índice

1. Sección introductoria	2
2. Análisis del problema	2
2.1. Consideraciones iniciales	3
2.2. Clases implementadas	3
3. Tareas para la solución del problema	3
4. Algoritmo diseñado	4

1. Sección introductoria

Este informe se hace con la intención de mostrar el análisis y diseño del problema planteado para el parcial 2 de la materia Informática 2. En él se podrán encontrar las tareas que parecen ser necesarias en la solución del problema, las consideraciones a tener en cuenta para lograr una implementación exitosa y el algoritmo que se diseñará sin tener en cuenta el lenguaje de programación y que servirá para la futura codificación del sistema solicitado.

2. Análisis del problema

En esta ocasión el problema planteado es hallar la forma de tomar una imagen cualquiera, en este caso las imágenes serían de banderas de los países del mundo, y procesarla de tal forma que los colores de esta imagen se vean representados en una matriz de LEDs que será simulada con ayuda de Tinkercad.

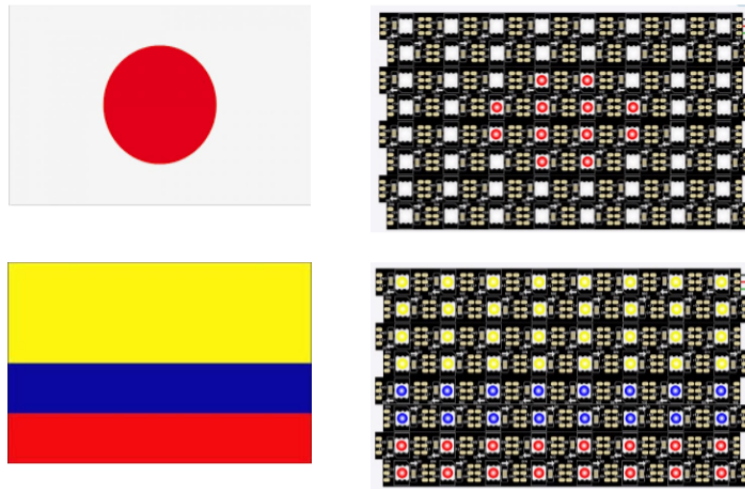


Figura 1: Ejemplo de representación.

El programa recibirá la ubicación de la imagen de la bandera, la cual podrá tener cualquier tamaño, y después de hacer el debido procesamiento de esta, tendrá como salida un archivo de texto en el que se escribirá el código que debe ser utilizado en el controlador de la matriz que será copiado y pegado del archivo txt al programa en Tinkercad y que finalmente, al simular, se vea representada la imagen deseada por el usuario.

Para este cometido se tendrán que utilizar los conceptos vistos hasta el momento, como el manejo de archivos para lectura y escritura, la creación de clases y objetos, la noción de herencia y el montaje de circuitos haciendo uso del Arduino en el programa de modelado 3D Tinkercad.

Teniendo en cuenta que es permitido usar la clase QImage, la cual se encarga de manipular y proporcionar acceso a los píxeles de cualquier imagen, permitiendo de esta forma conocer la composición del color de cada píxel de la imagen suministrada por el usuario utilizando el modelo RGB, el principal objetivo en este ejercicio es encontrar la forma de reducir o aumentar el tamaño de cualquier imagen al deseado (16x16) que es el tamaño que se escogió implementar para la matriz de LEDs.

Por último, es importante que el segmento de código necesario para rellenar la matriz de LEDs con los colores correspondientes a la imagen, y que es proporcionado por el programa implementado en Qt, sea escrito de una forma en la que sea fácil de copiar en el programa que controla la matriz, para así evitar errores en la simulación y que no sea posible visualizar los colores en los LEDs.

2.1. Consideraciones iniciales

Se debe tener en cuenta, según las restricciones, que no es permitido utilizar librerías para hacer el submuestreo o sobremuestreo de la imagen, se deberá desarrollar el código que nos permite minimizar o maximizar la imagen según lo requerido para que las nuevas matrices RGB tengan un tamaño de 16x16 y puedan ser mostradas en la matriz de LEDs montada en Tinkercad.

2.2. Clases implementadas

1. Clase MatrizRGB: Esta clase será la encargada de recibir la dirección de imagen ingresada por el usuario, y con está, obtener ancho y alto de la imagen y devolver una matriz `[3][ancho][alto]` que tendrá toda la información correspondiente a los componentes R, G, B de cada pixel de la imagen suministrada.
2. Clase newMatriz: Será la encargada de tomar los valores de cada pixel extraídos en las iteraciones sobre la MatrizRGB y según condicionales de saber si será submuestreo o sobremuestreo, irlos ingresando a una nueva matriz `[3][16][16]` que saldrá en un archivo .txt y contendrá la información de los componentes R, G,B para llevar al circuito de Tinkercad.

3. Tareas para la solución del problema

1. Entender el funcionamiento y las variedades de métodos de la clase QImage, la cual nos permitirá tener acceso a los pixeles de la imagen que queramos procesar, y de esta forma conocer la composición de color de estos.
2. Conocer el manejo de archivos haciendo uso de lenguaje C++, pues será necesario para la salida del programa que se va a crear para solucionar el problema.
3. Idear una forma de minimizar o maximizar el tamaño de la imagen original dependiendo de sus dimensiones para que se convierta en una imagen de 16x16. Los colores de la nueva imagen serán creados a partir de las tres matrices que nos son dadas por el modelo RGB de la imagen original, tomando cada una de estas y minimizándolas o maximizándolas según lo requerido.
4. Se tendrá que realizar el montaje de una matriz de LEDs de 16x16 en la plataforma de simulaciones de Tinkercad haciendo uso de las tiras de Neopixeles.
5. Se debe crear un código en Tinkercad que controle la matriz de LEDs y que además sea fácil de cambiar por el usuario, pues él deberá tomar la información dada por el programa en C++ y pegarla en cierta parte del código para que de esta forma, la implementación en Tinkercad, muestre los colores deseados en la matriz de LEDs.

4. Algoritmo diseñado

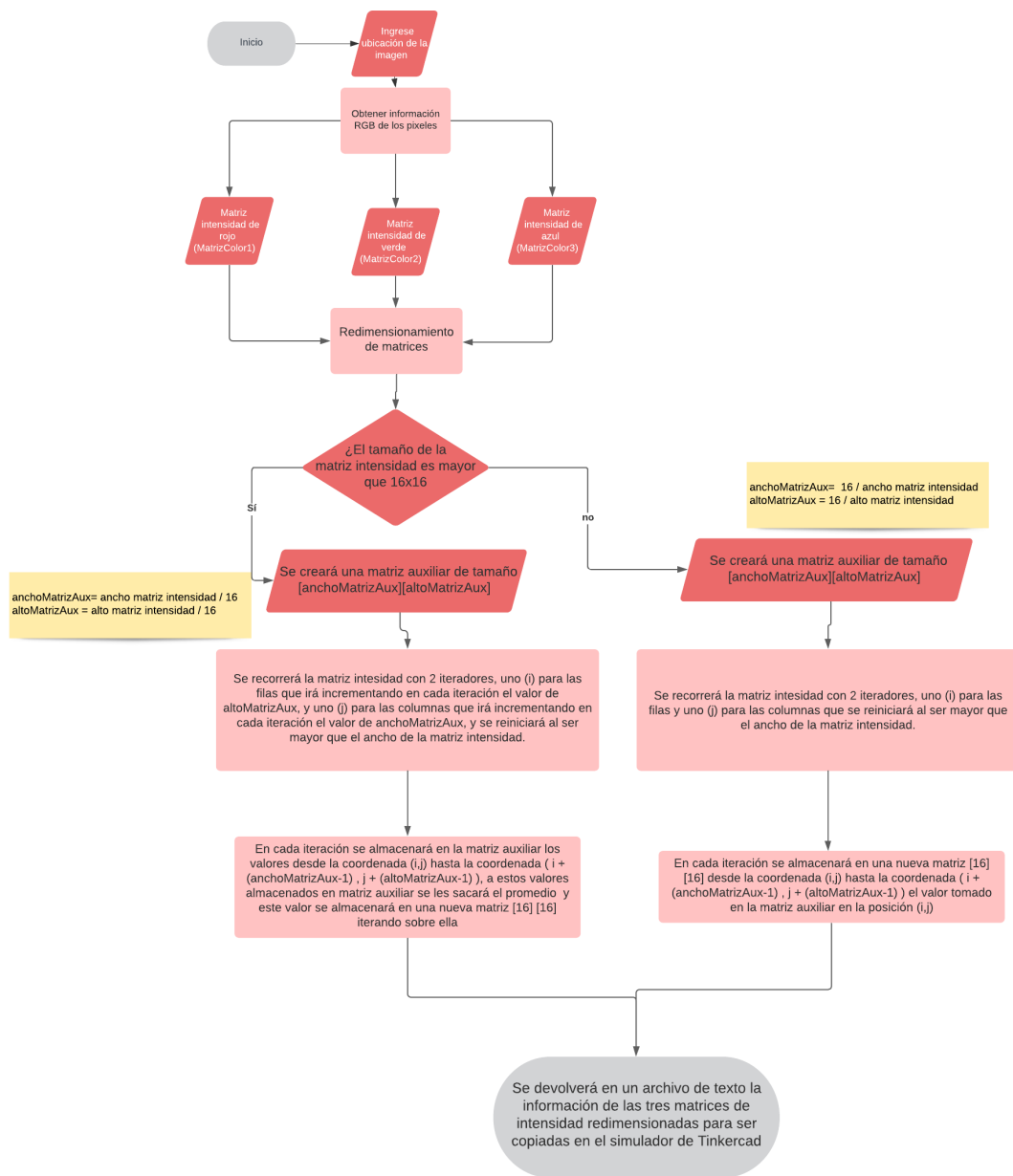


Figura 2: Algoritmo inicial.