

# **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

<b>1. Sección de contenido</b>	<b>2</b>
1.1. Solución del problema . . . . .	2
1.2. Clases implementadas . . . . .	2
1.3. Módulos de código implementados . . . . .	2
1.4. Esquema de la estructura final de las clases . . . . .	3
1.5. Circuito . . . . .	4
1.6. Problemas presentados . . . . .	4

# 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.

## 1.2. Clases implementadas

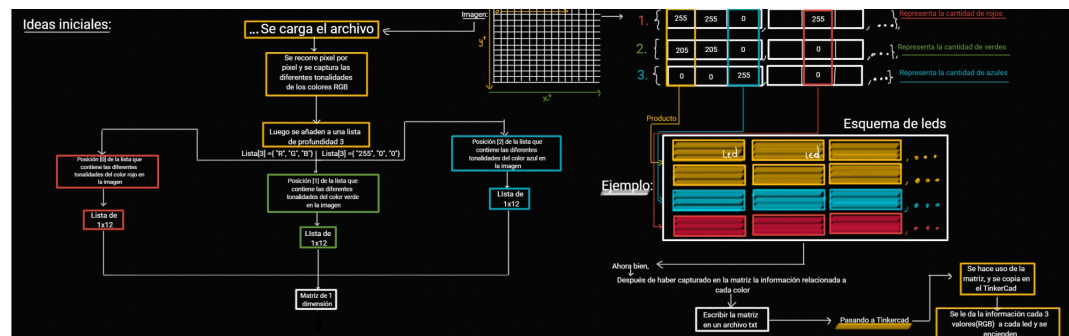
Para el funcionamiento del programa se creó una clase de nombre RGB, la cual se encarga de todo lo relacionado con el procesamiento de la imagen suministrada.

## 1.3. Módulos de código implementados

La clase RGB presenta las siguientes funciones públicas:

- CapturaRGB, la cual recibe como parámetro el objeto 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,

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.



cantidad de bloques

Ej:  $200 \times 200$

$\Rightarrow \text{tam} X = X / 12$

$\Rightarrow \text{tam} Y = Y / 12$

$\Rightarrow \text{Bloques} = (\text{tam} X) \times (\text{tam} Y)$

Para este ejemplo

Bloques  $50 \times 50$

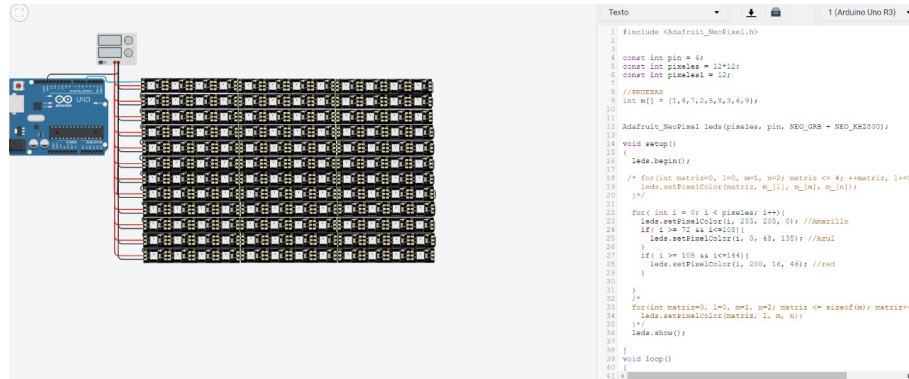
$\sum_{i=1}^{1500} \frac{R_1 + R_2 + \dots}{2500} = \begin{matrix} B_1, B_2, B_3, \dots, B_{12} \\ P_{1,0} \quad P_{1,1} \quad P_{1,2} \quad \dots \quad P_{1,11} \\ P_{2,0} \quad P_{2,1} \quad P_{2,2} \quad \dots \quad P_{2,11} \\ \vdots \\ P_{i,j} \quad \dots \quad P_{i,j} \end{matrix}$

$P_{0,0}$   
 $50 \times 50$

$0 + 1$


1
2
3
4
...

## 1.5. Circuito



## 1.6. 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.