

# **Informática II**

Parcial 2

**Juan Esteban Garcia Durango**  
**Jessica Valentina Gaviria Samboni**

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

# Índice

1. Prefacio	2
2. Clases Implementadas	2
3. Esquema donde describa la estructura final de las clases implementadas	2
4. Módulos de código implementado donde se evidencie la interacción entre las diferentes clases.	3
5. Estructura del circuito montado.	3
6. Problemas presentados	3

## 1. Prefacio

A continuacion se presenta la solución al desafío numero dos de la asignatura Informática 2, en cual se debe de representar una imagen de tamaño arbitrario en una matriz de neopixeles.

## 2. Clases Implementadas

El código cuenta con una única clase llamada Redimensionar”, que a su vez cuenta con 6 metodos, entre estos uno para submuestrear, otro para sobre-muestrear, y ademas uno que permite hacer ambos tipos de redimensionamiento .Estos posibilitan que se realice el ajuste de las medidas de la imagen dada por el usuario a ciertas medidas en las cuales la informacion pueda ser respresentada en la matriz de neopixeles de 8\*8.

## 3. Esquema donde describa la estructura final de las clases implementadas

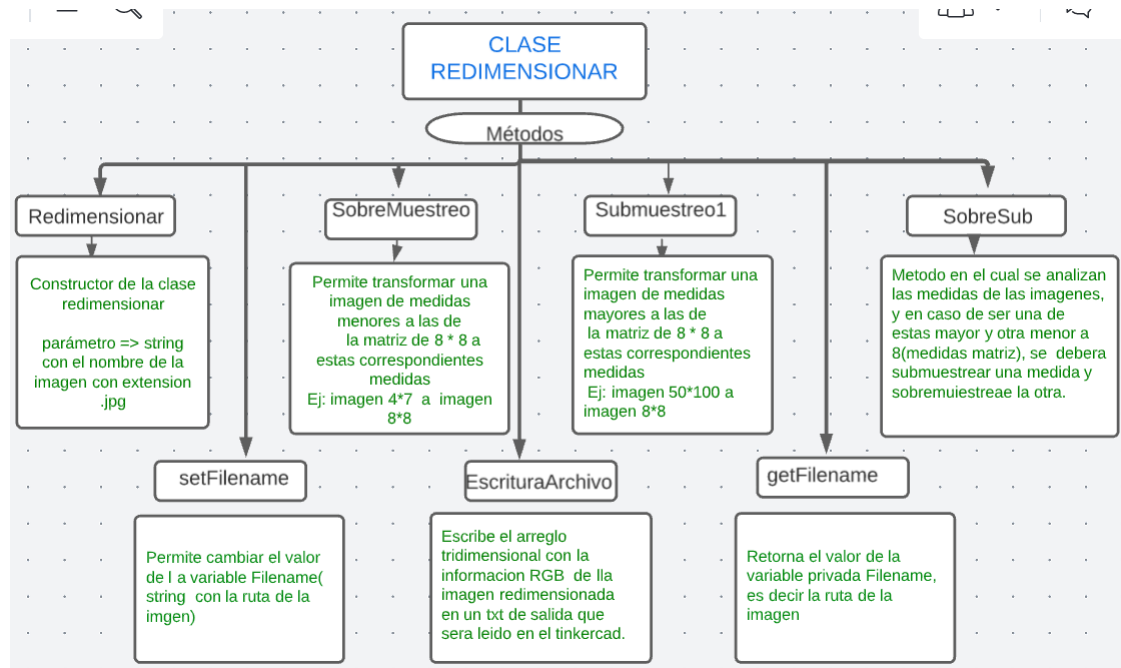


Figura 1: Clase redimension

#### 4. Módulos de código implementado donde se evidencie la interacción entre las diferentes clases.

Como anteriormente se mencionó, existe una única clase en el código, la cual contiene todos los métodos necesarios para los procesos de sobremuestreo y submuestreo, y adicionalmente la escritura en un txt de la información RGB de la imagen que posteriormente será leída en el tinkercad y representada en la matriz de neopixels.

#### 5. Estructura del circuito montado.

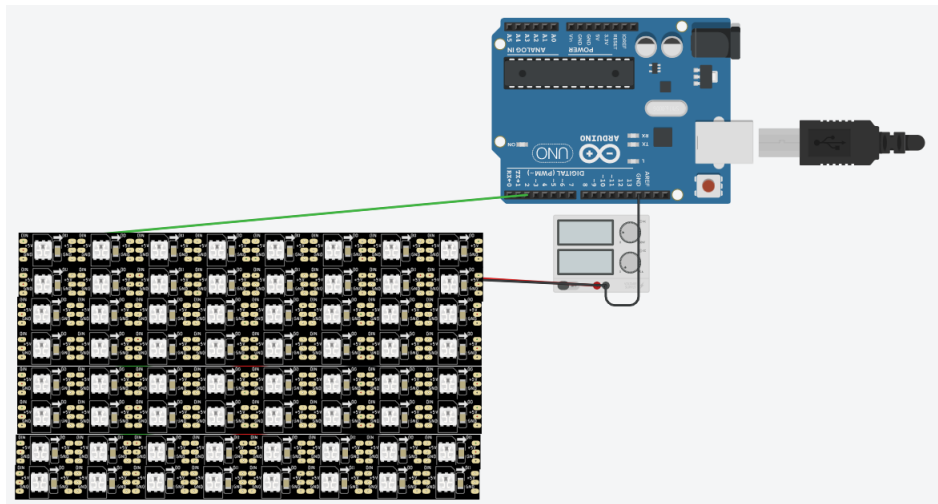


Figura 2: Montaje circuito

El circuito consta de 8 tiras de neopixeles, conectadas en serie, por lo cual están enlazadas a un único pin (pin Digital 2 en la figura 2). Cada tira a su vez cuenta con 8 Neopixels, a quienes se les ingresará la información RGB, guiando la asignación de cada uno de los valores RGB por un índice o posición.

#### 6. Problemas presentados

1. Uno de los problemas presentados fue el tener que hacer el montaje de nuevo de los neopixeles, ya que en primer lugar con una medida de 16\*16, el

programa no era capaz de correr.

2. Calculo de promedios en el submuestreo, acceder a posiciones especificas dentro de la imagen.

3. El metodo de submuestreo que elegimos, se basa en la idea de promediar grupos de pixeles de la imagen inicial, por lo cual se da el hecho de combinarse colores y en ocasiones no poder distinguirse un color en concreto.

4. Para el caso en el que se debe emplear submuestreo y sobremuestreo para una misma imagen, se hizo necesario emplear memoria dinamica a traves de contenedores, ya que el tamaño del arreglo que contiene la informacion RGB de los pixeles de la imagen, al ser el tamaño de uno de sus lados muy grande y el otro mas pequeño, es difícil manejar la cantidad de datos,