

# Implementación de la solucion

Juan Camilo Mazo Castro  
Santiago Pereira Ramirez

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

# Índice

1. Clases Implementadas	2
2. Esquema de la clase implementada	2
3. Modulacion	2
4. Estructura del circuito	3
5. Problemas presentados	4

## 1. Clases Implementadas

Para el desarrollo del parcial se creo una clase llamada imageresized al cual tiene atributos y metodos privados y metodos publicos los cuales proporcionan la funciones para el muestreo de la imagen insertada en el programa. Esta no tiene metodos sobrecargados o el constructor sobrecargado y utiliza algunas librerias como lo son string y fstream para el manejo de datos y llevarlos hacia un documento.txt, o tambien QImage para la extraccion de datos de la imagen ingresada.

## 2. Esquema de la clase implementada

En la Figura 1 se podra ver la estructura de la clase implementada en el entorno de desarrollo Qtcreator:

```
Librería #1
Librería #2
Librería #3
Librería #N //Librerías necesarias para el programa

Class Name//Nombre de la clase
{
    Atributos y métodos Privados: //Lo que debe de ser privado
        TipoDato Atributo #1;
        TipoDato Atributo #2;
        TipoDato Atributo #3;
        Void Metodo_1();
        Void Metodo_2();
        Void Metodo_3(TipoDato Matriz1, TipoDato Matriz2, TipoDato Matriz3);
        Void Método_N (){}
    Atributos y métodos públicos: //constructor y lo que debe de ser publico
        Name (TipoDato String_Name);
        Void Metodo_1();
}
```

Figura 1: Estructura clase

## 3. Modulacion

En la figura 2 se podra ver la modulacion de la clase Imageresized:

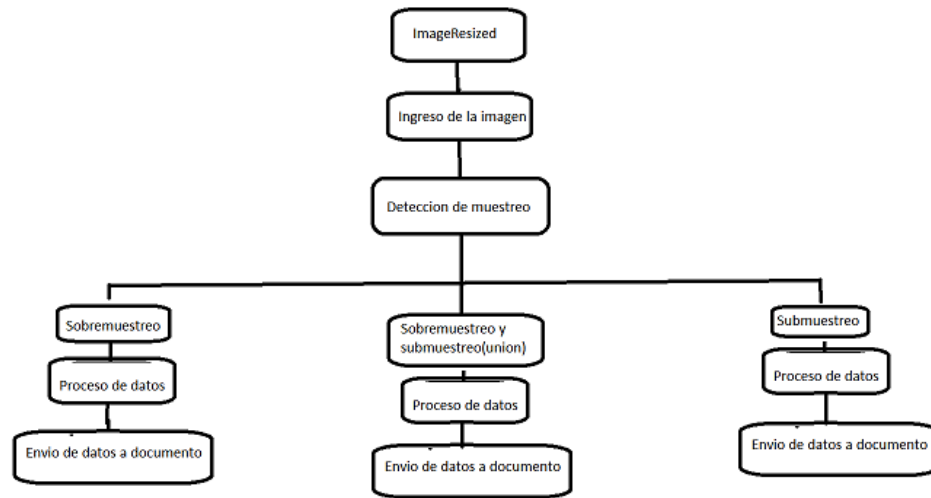


Figura 2: Esquema de la clase

## 4. Estructura del circuito

Para la construccion del circuito se utilizaron elementos como una placa electronica arduino, un suministro de energia y 16 tiras de leds neopixel. Primero por medio del pin 2 del arduino lo conectamos a la entrada de la primera tira de neopixel, despues atravez del GND(tierra) del arduino lo conectamos en la tierra del suministro de energia y este a su vez a la tierra de la primera tirilla de neopixel. Respecto a la potencia se empalmo directamente del suministro de energia a la potencia de la primera tira de neopixel.

Ahora por medio de la salida de la primera tirilla de neopixel se conectara a la entrada de la siguiente tirilla de neopixel, al igual que la potencia y la tierra repitiendo este proceso hasta terminar con las tiras.

A continuacion se podra ver un pequeño ejemplo en la figura 3.

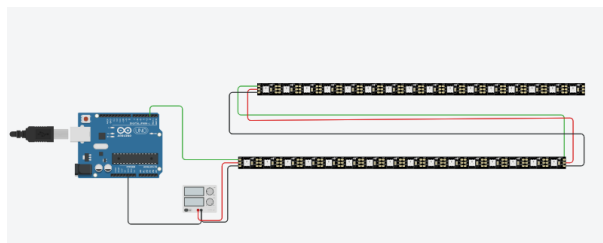


Figura 3: Ejemplo estructura circuito

## 5. Problemas presentados

Durante el desarrollo del programa y montaje del circuito se presentaron alguno problemas como puede ser:

–Al utilizar tres arreglos para almacenar los datos en tinkercad la ejecucion y en general para mostrar la bandera se tardaba bastante tiempo, para solucionarlo se debio de utilizar un arreglo tridimensional y (for) anidados para mandar los datos.

–Al ejecutar el programa en tinkercad seguia tardando mucho, esto ya que era problema del propio computador y conexion a internet de un participante, por el contrario el del otro compañero se ejecutaba rapido y eficaz. Esto ocasiono que algunas veces se cambiara el Hardware, pero al final no hubo una necesidad de cambiar la estructura inicial del mismo.

–En el programa se trabajo especialmente con arreglos, esto ocasiono que al ingresar imagenes muy grandes se llenara el stack, para dar solucion a esto se recurrio a memoria dinamica.

–Se tuvieron algunos inconvenientes con el Github y los repositorios, ya que se clono mal en un caso y ademas el enlace avaces tardaba mucho entre el repositorio remoto y local.