

# Informe analisis y diseño - Parcial 2 - Informatica II

Daniel Andres Agudelo Garcia

Andres Felipe Rendon

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

# Índice

<b>1. Problemas presentados y trabajo que se fue haciendo</b>	<b>2</b>
<b>2. Analisis y diseño del parcial</b>	<b>3</b>
2.1. Primeras ideas e impresiones . . . . .	3
2.2. Esquema . . . . .	4
2.3. Algoritmo implementado . . . . .	5
<b>3. Consideraciones a tener en cuenta</b>	<b>6</b>
<b>4. Referencias usadas</b>	<b>6</b>

## 1. Problemas presentados y trabajo que se fue haciendo

Primero necesitamos investigar y tener amplio conocimiento de como implementar el codigo para la redimension de las imagenes, ya que el montaje de arduino es mas sistema electrico y podemos pedir asesoria en la ayuda de este.

Los primeros dias es investigacion del manejo y utilizacion de imagenes en C++.

Tuvimos problemas con el codigo de QT en la lectura de la imagen, la cual se tuvo que corregir cierto detalles menores ya que la imagen generaba ¡not accessible¿por error en la indicacion de la ruta, pero fue solucionado.

Se investiga el funcionamiento de QImage, ya que tiene mucha funcionabilidad, buscando como redimesionar las imagenes.

Se investigo la convolucion, la cual nos permitiria hacer edits a las imagenes, es una posible solucion a la problematica, de igual forma no se sabe si será el metodo a implementar, es como aplicar cierto filtro, pero veremos si puede ser la solucion necesaria para el problema.

## 2. Analisis y diseño del parcial

### 2.1. Primeras ideas e impresiones

Obervamos la problematica presentada, analisando punto por punto del trabajo presentado, señanalandando la parte mas complicada que seria el algoritmo a implementar en QT, investigando por diferentes paginas como cambiar las dimensiones de una imagen, ( mas abajo se adjunta las paginas que fueron consultadas.

Se tiene como primera idea, la creacion de 3 clases, la primera que lea la imagen que el usuario ingreso, la segunda que redimensione la imagen, y la tercera impresion ordenada.

Se piensa en la implementacion de dos funciones que se escriben por separado y en otra funcion usarlas, que seria el uso de el ancho y el alto de la imagen.

La idea es usar un objeto que herede a QImage, y demas objetos, en el cual alli se usen dos funciones, el ancho y el alto, para la redimension de una imagen que se ingrese y dejarla en tamano estandar, por ejemplo si la imagen es de tamano 800x800 dejarla en 200x200, o si es 150x150 convertirla a 200x200, y ya mas adelante convertir esta imagen de 200x200 en una de 16x16 que es son los leds a utilizar en tinkercad.

Usamos el metodo scale de la libreria QImage, ya redimensionamos la imagen en una matriz de 16x16 donde nos muestra la intensidad de color de cada pixel de los 16x16.

Una opcion encontrada es tomar la imagen que nos haya señalado el usuario y hacer una copia de esta, cambiando la dimension por la requerida que serian 16x16.

## 2.2. Esquema

Se piensan usar 3 clases, que tengan varias funciones, las cuales serian,

1. Clase donde se ingrese la direccion o nombre de la imagen que se desea mostrar, tendra una funcion que ingresara el string en donde leera y guarde o la copie, trabajando mas adelante con la copia de esta imagen guardandola en cierta variable o espacio de la memoria.

2. Clase donde se redimensionara la imagen leida, en una matriz de 16x16 que es lo que trabajaremos en tinkercad, teneiendo la imagen cargada en una variable o en la memoria, tendremos las funciones donde se encargue de recorrer la imagen y la conviertan en una matriz 16x16 con sus respectivos colores con sus colores rojo,verde,azul.

3. Despues de tener guardado los 16x16 pixeles que se guardaron en la matriz con sus colores rojo,verde,azul, se creara un archivo/variable donde se organiza todos los datos guardados y los genere en un .txt para que sea solo copiarlos y pegarlos en el tinkercad como dice las instrucciones.

## 2.3. Algoritmo implementado

Vamos a pedirle al usuario que ingrese el nombre de la imagen que desea leer, y como estamos manejando banderas se le preguntara al usuario con que bandera de desea trabajar, por ejemplo que ingrese brasil, entonces se sabe que trabajaremos con la bandera de brasil . O en interfaz grafica tener un boton que lo lleve a los archivos donde el usuario directamente señalara la imagen con la que desea trabajar.

Al señalar la imagen con la que trabajaremos se hara una copia en la que se guardara en una variable/matriz/contenedor/archivo, que la mande a una funcion/clase que la lea y sepa cual es su alto y ancho.

Una funcion/clase que recorra un bloque de pixel y lo vaya guardando en una matriz, por ejemplo si la imagen es de 600x600 que recorra la imagen y cada 40 pixeles guarde uno en la variable/matriz/contenedor/archivo que mas adelante sera la cual muestre

Despues de tener una imagen de tamaño 15x15, guardar los datos ordenadamente en un variable/matriz/contenedor/archivo separando esto por los colores rojo, azul y verde, haciendo de una imagen se generen 3 cadena de numeros que representen la intensidad del color.

Generar un archivo .txt donde muestre ordendamente la cadena de valores que representan los colores de la imagen para que el usuario solo sea copiarlo y pegarlo en el monitor de tinkercad

En tinkercad se implemento una matriz 16x16 de leds, los cuales mostraran los colores que ingrese la cadena de valores que copio el usuario en el programa de QT, lo hara para la intensidad de los 3 colores, azul, rojo y verde, y ya la matriz de leds mostrara la imagen que desde un principio el usuario ingreso.

### 3. Consideraciones a tener en cuenta

### 4. Referencias usadas

<https://es.stackoverflow.com/questions/182175/redimensionar-imagen-con-qpixmap>

<https://www.programcreek.com/cpp/?CodeExample=resize+image>

<http://www.cplusplus.com/forum/general/2615/>

<https://doc.qt.io/qt-5/qimage.html.image-transformations>

<https://cpp.hotexamples.com/examples/png/image/resize/cpp-image-resize-method-examples.html>

<https://www.youtube.com/watch?v=HGHbcRscFsg>

<https://www.youtube.com/watch?v=fbH005SzEMc>

<https://www.youtube.com/watch?v=028GNYC32Rg>

<https://www.youtube.com/watch?v=6byB1jX3Fnk>

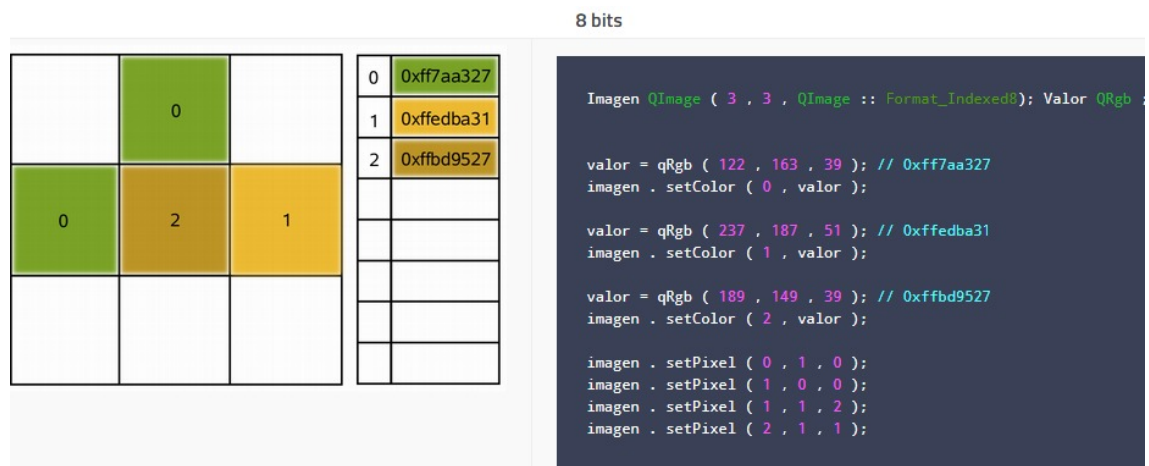


Figura 1: Ejemplo uso QImage

Función	Descripción
<code>setDotsPerMeterX()</code>	Define la relación de aspecto estableciendo el número de píxeles que encajan horizontalmente en un medidor físico.
<code>setDotsPerMeterY()</code>	Define la relación de aspecto estableciendo el número de píxeles que caben verticalmente en un medidor físico.
<code>llenar()</code>	Rellena toda la imagen con el valor de píxel dado.
<code>invertPixels()</code>	Invierte todos los valores de píxeles en la imagen usando el valor <code>InvertMode</code> dado .
<code>setColorTable()</code>	Establece la tabla de colores utilizada para traducir índices de color. Solo formatos monocromáticos y de 8 bits.
<code>setColorCount()</code>	Cambia el tamaño de la tabla de colores. Solo formatos monocromáticos y de 8 bits.

Figura 2: Funciones de QImage