



**PROYECTO**  
**FILTROS DE IMAGEN**

**Presentado a:**  
**JEFFERSON AMADO**

**Integrantes:**

<b>INGRID ECHEVERRY MONTOYA</b>	<b>1943542</b>
<b>JEAN PIE CARDENA PEREA</b>	<b>1942703</b>
<b>ANDRE DAVINCHO CAMARON</b>	<b>1944140</b>
<b>JHAN ALEJANDRO PEREZ UMBARILA</b>	<b>1941003</b>

**ESCUELA DE INGENIERÍA INGENIERIA DE SISTEMAS Y COMPUTACIÓN**  
**FUNDAMENTOS DE PROGRAMACIÓN**  
**UNIVERSIDAD DEL VALLE**  
**SANTIAGO DE CALI**  
**2020**

## TABLA DE CONTENIDO

---

<b>Resumen .....</b>	<b>3</b>
<b>Introducción .....</b>	<b>3</b>
<b>Objetivo General.....</b>	<b>3</b>
<b>Objetivos específicos.....</b>	<b>4</b>
<b>Argumentos .....</b>	<b>4</b>
<b>Glosario.....</b>	<b>4</b>
<b>Conceptos .....</b>	<b>5</b>
<b>Filtros .....</b>	<b>5</b>
<b>QT Creator .....</b>	<b>5</b>
<b>POO (Programación orientada a objetos) .....</b>	<b>5</b>
<b>C++ .....</b>	<b>5</b>
<b>Encapsulamiento .....</b>	<b>5</b>
<b>Polimorfismo .....</b>	<b>5</b>
<b>Herencia .....</b>	<b>5</b>
<b>Abstracción.....</b>	<b>5</b>
<b>Diseño del programa .....</b>	<b>6</b>
<b>Manual uso del programa .....</b>	<b>7</b>
<b>Visualización de los filtros .....</b>	<b>8</b>

## **Resumen**

Para este proyecto se elaboró un programa capaz de aplicar filtros a imágenes que se pueden leer desde su propio archivo de texto para manipularlas logrando distintos efectos en estas. Para lo cual se utilizaron las herramientas aprendidas en clase de programación orientada a objetos, en el lenguaje C++ y con ayuda de QT creator para crear la interfaz gráfica.

El proyecto fue escrito haciendo uso del paradigma orientado a objetos aplicando sus 4 conceptos fundamentales.

**PALABRAS CLAVE:** Abstracción, polimorfismo, herencia y encapsulamiento.

## **Introducción**

Image processing es un programa que permite manipular imágenes para aplicarles filtros. Entre estos filtros se pueden encontrar para convertir una imagen a blanco y negro, añadirle un tono sepia, colorearla, girarla 90 grados, invertirla, sobreponer imágenes e incluso limpiar su fondo con tonalidades verdes indeseadas. Un ejemplo de estos programas es el conocido Photoshop que se creó en el 1990 y a día de hoy a seguido actualizándose añadiendo nuevas características para la manipulación de imágenes. Además hemos ofrecido al usuario la oportunidad de usar las herramientas de una manera que le resulte amable y intuitivo gracias a una interfaz gráfica diseñada para facilitar el uso de estas mismas.

En este informe se describe el proceso de diseño y construcción para nuestro programa de procesamiento de imágenes, explicando detalladamente cada parte detrás del desarrollo y programación del proyecto. Utilizando los conocimientos adquiridos en las clases de POO y por investigación propia.

## **Objetivo General**

Asimilar el conocimiento adquirido durante el curso para crear un programa que implemente los 4 pilares fundamentales de la programación orientada a objetos.

## **Objetivos específicos**

- Trabajar en comunidad para aprender a leer código y documentarlo correctamente para facilitar la colaboración unánime.
- Optimizar el programa con herramientas como la herencia y el encapsulamiento.
- Hacer uso de la modularidad para operar desde clases que trabajaran en conjunto.
- Desarrollar un programa de procesamiento digital de imágenes con los requerimientos que el profesor exige, y aprendiendo un poco sobre los algoritmos que el proyecto requiere.

## **Argumentos**

Los programas de procesamiento de imágenes han sido de gran utilidad a lo largo de los años para labores como el diseño publicitario, graficas estadísticos, entretenimiento, diseño gráfico y entre muchas otras.

Nuestro proyecto consiste en la implementación de algunos filtros como el colorice que se encargara de resaltar tonalidades de imágenes blanco y negro haciendo que se nos permita ver una idea de la intensidad de los colores de esta misma, un colorToGrey que convertirá una imagen de color en una de tonos negros y blancos, un sepia que repartirá tonos en una escala de marrones, un painting que dara a elegir entre tres tonalidades, un flip que podra voltear la imagen a 90 grados, un mirror que invertirá la imagen y un filtro negative que dara un efecto de colores negativos. Hemos añadido, además, una interfaz con botones sencillos que el usuario usara sin dificultad para sacar provecho de las herramientas que le ofrecemos.

## **Glosario**

### **Procesamiento digital de imágenes**

El procesamiento de imágenes digitales es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

## **Conceptos**

### **Filtros**

Se trata de métodos para resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o también para ocultar valores anómalos.

### **QT Creator**

Es un IDE multiplataforma programado en C++, JavaScript y QML creado por Trolltech para el desarrollo de aplicaciones con Interfaces Gráficas de Usuario con las bibliotecas Qt.

### **POO (Programación orientada a objetos)**

Es un paradigma de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basada en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

### **C++**

Es un lenguaje enfocado al paradigma orientado a objetos que extendió las funcionalidades de su antecesor C para implementar mecanismos que permitieran la manipulación de objetos.

### **Encapsulamiento**

Consiste en poder ocultar información del estado de un objeto que al usuario no se le es necesaria.

### **Polimorfismo**

Consiste en la sobrecarga de métodos y la sobreescritura de estos mismos logrando tener distintas funcionalidades en uno mismo.

### **Herencia**

Consiste en el poder de pasar los métodos y atributos de una clase a otra para poder ser reutilizados.

### **Abstracción**

Es la capacidad para extraer las características esenciales de un objeto separando las que se pueden generalizar en la unidad más mínima posible.

## Diseño del programa

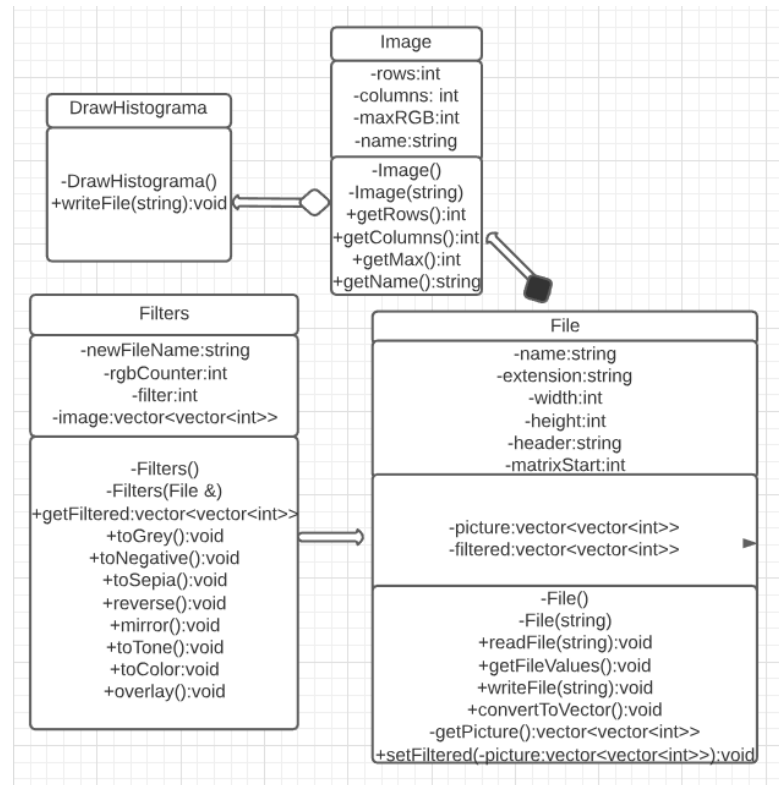


Figura 1. Diagrama UML

El desarrollo del programa comienza definiendo el IDE que se usará, como podrá ser el caso de QT CREATOR o de el entorno de desarrollo repl.it y decidiendo apoyarnos en la app de irfanview para obtener el formato PPM y PGM que permiten editar las imágenes desde una matriz RGB; el paradigma a cumplir será el de orientado a objetos.

En los primeros días discutimos sobre nuestros enfoques, como se procesaría la lectura y escritura de el archivo que contenga la imagen y entonces proseguimos a ahondar en los filtros y decidir la repartición de los mismos.

Habrà una interfaz con la que el usuario podrá interactuar de forma intuitiva seleccionando la imagen deseada y apareciéndole a su lado el cambio que desee realizarle.

Para el procesamiento de lectura del archivo matriz de la imagen, usamos las librerías de ifstream y ofstream, que permiten capturar las líneas de una hoja de texto, con lo que implementamos un algoritmo donde convertiríamos a números cada string del texto procediendo a guardarlos en un array bidimensional. Descubrimos que en algunos filtros habia un problema, cada que manipulábamos los valores obteníamos cambios no deseados porque influíamos directamente en el arreglo de la imagen original, por lo que hicimos un vector de vectores que funcionaria como la copia de esta dándonos la libertad de manipular la original con los valores preestablecidos sin alterar los resultados deseados, para esto se utilizo la

herencia ya que heredaríamos métodos similares a los del array original aprovechando que el vector sería una imagen, una vez tenido en la clase File se leían con el método de writeFile que tendría a la imagen original para extraer sus valores ya definidos en el array respectivo. Luego creamos otra clase dedicada a los filtros que tendría a la imagen y al vector copia, comenzamos por los algoritmos que tenían relación con los tonos de colores como el caso de colorToGrey que tiene la siguiente fórmula:

$$\text{gray} = 0.2989 * \text{red} + 0.5870 * \text{green} + 0.1140 * \text{blue}$$

Esta fórmula consiste en multiplicar un número dependiendo del tono. Siguiendo el mismo tipo de algoritmo se realizó el colorice que contiene 3 tipos de colores para pintar la imagen, rosa, verde y azul y el negative. En grayToColor para resaltar las tonalidades de una imagen gris y detectar cuáles eran más fuertes o débiles, a cada R, G Y B le manipulamos para acrecentar sus valores hasta volver a tener una tonalidad que resaltara sus puntos de más intensidad.

En el flip intercambiamos las columnas con las filas y en el sobreponer elegimos la imagen más grande para que la de menor medida se rellene con color verde hasta adquirir el tamaño de su fondo, para luego proceder a eliminar el verde teniendo como resultado la imagen recortada sobre otra imagen.

Entonces se requirió hacer un histograma que escalaría toda la imagen para ver cuántas tonalidades tenía cada color del 0 al 255, el histograma tuvo que realizarse en una clase aparte pues debía de escribir un archivo, por lo que era parecido a la clase File que se encargaba de la escritura y lectura, solo que esta vez no se haría uso de la lectura y el método de escritura se sobrescribiría para adecuarlo a escribir la lista de los valores del histograma.

En el flip intercambiamos las columnas con las filas y en el sobreponer elegimos la imagen más grande para que la de menor medida se rellene con color verde hasta adquirir el tamaño de su fondo, para luego proceder a eliminar el verde teniendo como resultado la imagen recortada sobre otra imagen.

### **Manual uso del programa**

El usuario tendrá un menú de interfaz amigable con la que podrá interactuar y moverse a través de distintos botones y opciones que aparecen en la pantalla.

Click here to continue: Será el botón de inicio que te abrirá la ventana donde podrá empezar a usar el programa.













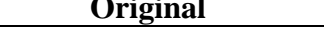
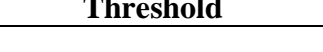
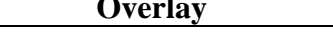
Cargar imagen: Te permite seleccionar una imagen de entre tus archivos para mostrarla en un cuadro apartado a la izquierda.

Filtro: En el centro abra un botón desplegable que contendrá los filtros que poseemos actualmente, al seleccionar uno se apreciará el efecto en el cuadro apartado en la derecha.

Guardar: Permite alojar el efecto de la imagen realizado entre tus archivos.

## Visualización de los filtros

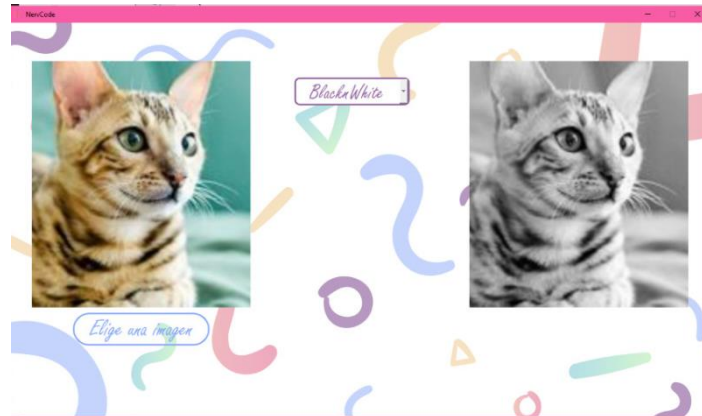
Actualmente contamos con los siguientes filtros:

Original	Black and White	Negative
		
Original	Sepia	Tone
		
Original	Tone	Tone
		
Original	Gray to color	Threshold
		
Original	Threshold	Overlay
		



## Resultados:

El proyecto se realizó de forma exitosa manteniendo la promesa del paradigma orientado a objetos y los filtros requeridos contando con una interfaz gráfica con la que el usuario puede interactuar.



*Figura 2 Menú inicial del programa*

Se consiguió implementar los conceptos vistos en clase como herencia, encapsulamiento y el uso de punteros, recurriendo al Qt creator para realizar una interfaz gráfica.

Finalmente se logró crear un programa que procesa imágenes y aplica efectos sobre estas.

## Bibliografía

1. 4rsoluciones. *¿Qué son los paradigmas de programación?* (online). 2013. Disponible en: <https://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>
2. <https://gavilanch.wordpress.com/2018/07/05/los-4-pilares-de-la-programacion-orientada-a-objetos/>
3. [https://es.wikipedia.org/wiki/Procesamiento\\_digital\\_de\\_im%C3%A1genes](https://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes)
4. <https://www.qt.io/download>
5. [https://es.wikipedia.org/wiki/Procesamiento\\_digital\\_de\\_im%C3%A1genes](https://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes)