

Instituto Politécnico Nacional

Escuela Superior de Computo

3CV16

Aplicaciones para comunicaciones en red

Badillo de la Rosa Jorge Antonio.

Gutiérrez Rodríguez Omar

Reyes Pérez Edward Daniel

Practica 1:   
Aplicación de filtro RGB a imágenes.

Índice

[Introducción. 3](#_Toc112887798)

[Desarrollo. 4](#_Toc112887799)

[Diagrama de flujo. 4](#_Toc112887800)

[Explicación. 5](#_Toc112887801)

[Pruebas. 7](#_Toc112887802)

[Conclusiones. 9](#_Toc112887803)

# Introducción.

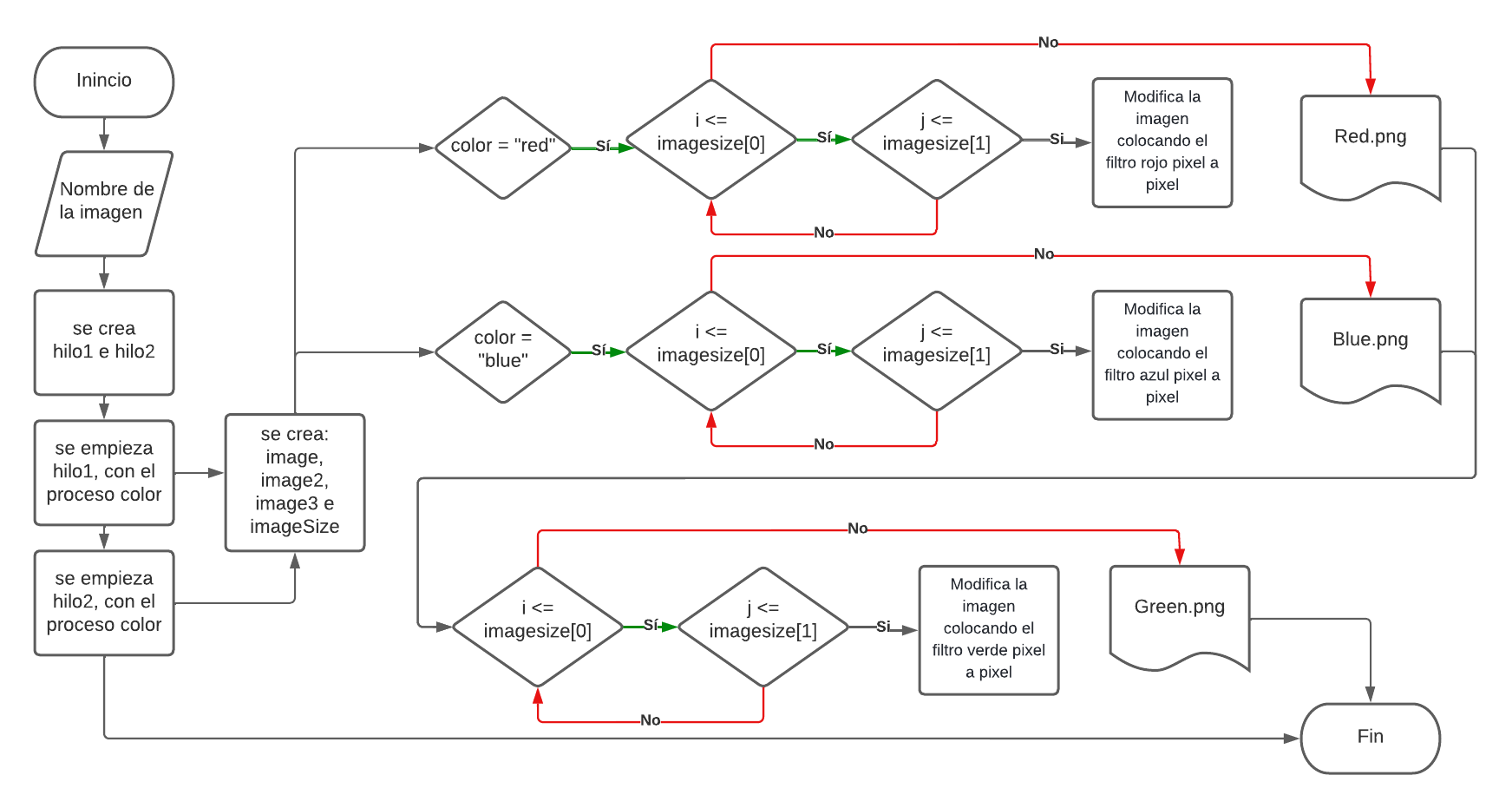
En esta práctica aplicaremos los filtros RGB a una imagen que cargaremos en el programa. El programa deberá generar tres imágenes (una para cada filtro; rojo, verde y azul).

El programa ejecutará dos subprocesos (hilos) donde se aplicará el filtro y se genera la imagen, la imagen “roja” se genera con el primer hilo, la imagen “azul” con el segundo hilo, mientras que la imagen “verde” se genera con ambos hilos.

Para desarrollar la práctica utilizamos el lenguaje de programación Python, utilizamos las librerías cv2; para manjar imágenes, y Thread; para manejar hilos.

# Desarrollo.

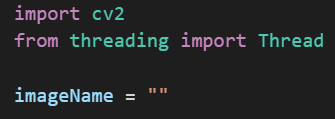
## Diagrama de flujo.



## Explicación.

Se impartan las librerías que se ocupan en la práctica, cv2; para manejar imágenes, y Thread; para manejar hilos.

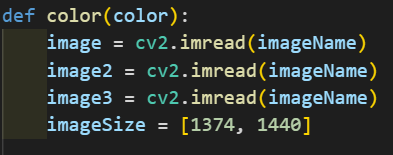
Se declara la variable imegeName, donde se guarda el archivo al que le aplicaremos filtros.



Se declara la función color(), que será la función que realizara el hilo. El argumento color, nos servirá para ejecutar el segmento de código que realizará cada hilo.

El método cv2.imread(imageName) cargara la imagen y retorna la imagen en forma de matriz. Se carga la imagen en tres variables distintas, para aplicar los tres filtros.

Se declara un arreglo con el valor de las filas y columnas.

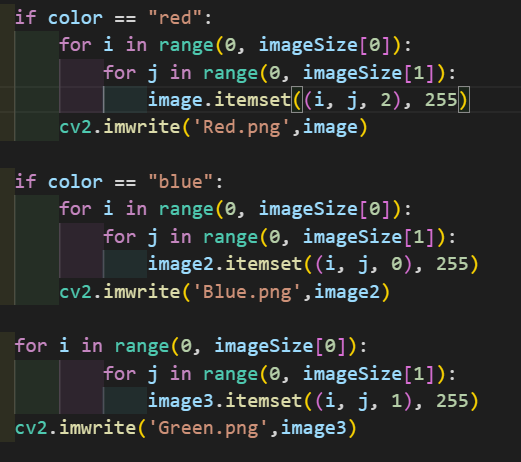


Se segmenta la función utilizando condicionales con la finalidad de aplicar el filtro dependiendo del hilo a ejecutar.

El método itemset( (fila, columna, capaColor), valorColor), nos permite modificar el valor del pixel. El valor máximo de valorColor = 255, el valor de la capaColor = 0 (azul), 1 (verde) y 2 (rojo).

El método cv2.imwrite(nombreArchivo, imagen) guarda una imagen en el directoria de trabajo actual.

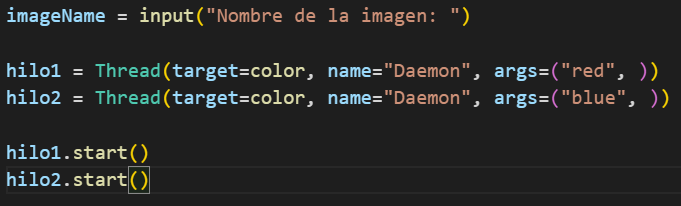
Usando ciclos anidados, se modifica pixel a pixel se va aplicando el filtro. El primer ciclo recorre las filas y el segundo las columnas.



El método Thread(funcionEjecutar, nombreHilo, arguemtos) crea el hilo.

El método .start() inicia el hilo.

Después que el programa reciba el nombre del archivo (imagen), se crean dos hilos, para ser ejecutados posteriormente.



# Pruebas.

A continuación, se muestra la carpeta que contiene los archivos necesarios para realizar la práctica.

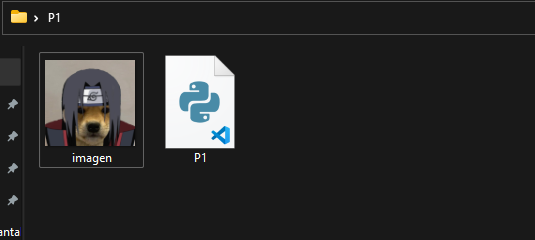


Ilustración : Carpeta sin ejecutar el programa.

A continuación, se muestra la terminal.



Ilustración : Terminal.

A continuación, se muestran la carpeta, después de haber ejecutado el programa. Donde se encuentra el resultado de aplicar los filtros RGB.

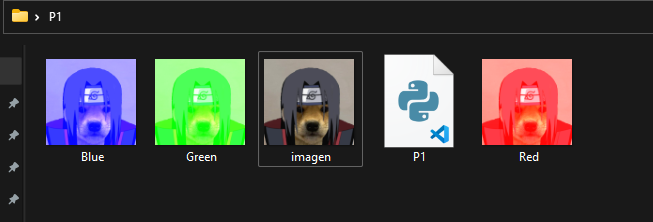


Ilustración : Carpeta después de ejecutar el programa.

A continuación, mostraremos la imagen original y el resultado de aplicar los filtros.



# Conclusiones.

Cuando utilizamos hilos, estos se ejecutan de manera simultanea entre ellos y al proceso principal. Los hilos permiten ejecutar “funciones”, mientras el proceso principal se sigue ejecutando, debido a que ejecutan una función, es posible realizar tareas similares sin necesidad de escribir más código (como fue el caso de aplicar los filtros, dependiendo del argumento que enviamos en la función.

En Python se maneja el sistema BGR (Blue, Green, Red), se maneja utilizando capas de color, donde 0 (azul), 1 (verde) y 2 (rojo). Estos valores pueden tomar un valor mínimo de 0 y un máximo de 255.

En la práctica también podemos comprobar que las imágenes se manejan como matrices de pixeles (fila, columna, capa de color).