

En este proyecto se busca los estudiantes hagan uso de la teoría de programación orientada a objetos para gestionar y modificar imágenes.

1. Escribir un clase que reciba una imagen en formato RGB ($n \times m \times 3$). El constructor debe almacenar la cantidad de filas (alto) y columnas (ancho) de la imagen.
2. Escribir un método que reciba una letra, 'R', 'G' o 'B', y visualice la componente R, G o B de la imagen en el mapa de colores correspondiente. <https://matplotlib.org/stable/tutorials/colors/colormaps.html>.
3. Escribir un método que permita enfocarse en un área rectangular de la imagen. Debe recibir fila y columna inicial, y fila y columna final. Cualquier modificación que se realice sobre la imagen, se realizara sobre el área enfocada solamente. El enfoque se activa con un método de activación, y se desactiva con un método de des-activación.
4. Escribir un método que reciba un vector o lista de 3 elementos $[\alpha_R, \alpha_G, \alpha_B]$, y modifique los valores de la matrices de esta manera:

$$\mathbf{R}_{\text{new}} = \alpha_R \mathbf{R}, \quad \mathbf{G}_{\text{new}} = \alpha_G \mathbf{G}, \quad \mathbf{B}_{\text{new}} = \alpha_B \mathbf{B},$$

Donde \mathbf{R} , \mathbf{G} y \mathbf{B} son las matrices de cada canal de color. El método modifica la imagen con los nuevos valores \mathbf{R}_{new} , \mathbf{G}_{new} y \mathbf{B}_{new} .

5. Escribir un método que filtre un canal en especial de acuerdo a un valor de umbral. Por ejemplo sí el usuario invoca el método con 50 y 'R', se hacen 0 todos valores del canal R menores a 50.
6. Sobrecargar el operador suma de tal manera que se pueda sumar una lista con 3 elementos $[\beta_R, \beta_G, \beta_B]$ con cada una de la matrices R, G y B. Tener en cuenta si esta activado el enfoque.
7. Escribir métodos que permitan visualizar la imagen completa y la parte enfocada.
8. Escribir métodos que permitan guardar la imagen completa y la parte enfocada.
9. Escribir un código que permita verificar el correcto funcionamiento de la clase.