

Algoritmo 3

Muñoz Nuñez Ian Emmanuel

Negativo de la imagen

Visión Robótica

Ejercicio 1

```
# Se importa la libreria opencv para abrir y manipular la imagen
import cv2

# Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
imagen = cv2.imread("rgb.jpg")

# Se genera una imagen en escala de grises
grises = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)

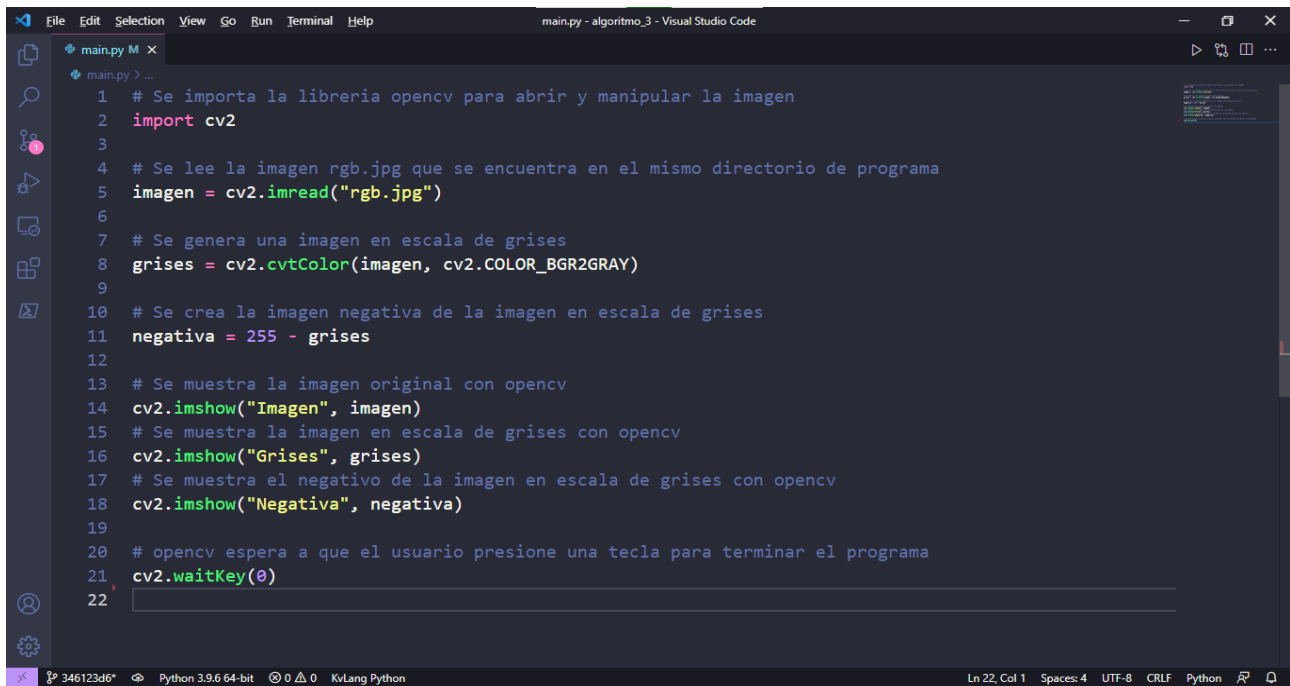
# Se crea la imagen negativa de la imagen en escala de grises
negativa = 255 - grises

# Se muestra la imagen original con opencv
cv2.imshow("Imagen", imagen)
# Se muestra la imagen en escala de grises con opencv
cv2.imshow("Grises", grises)
# Se muestra el negativo de la imagen en escala de grises con opencv
cv2.imshow("Negativa", negativa)

# opencv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

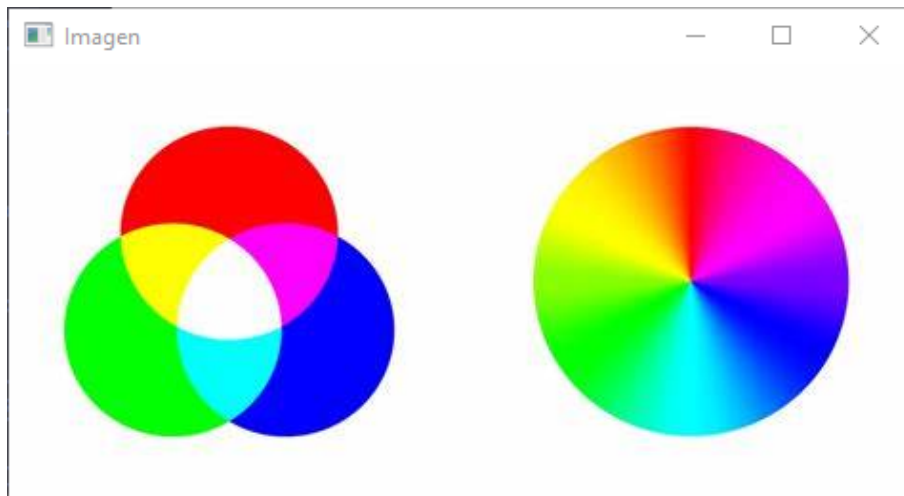
Para el código del ejercicio 1 primero se importa la librería opencv, pues se necesita para leer la imagen y manipularla. Después se carga la imagen "rgb.jpg", se declara la variable "grises" que es igual a la imagen original pero con escala de grises, luego se genera la variable "negativa", que es el resultado de convertir a negativo la imagen en escala de grises, por último se muestran la imagen original, la de escala de grises y la negativa. Al final opencv espera a que el usuario presione una tecla para terminar el programa.

Imagen del código del ejercicio 1



```
1 # Se importa la libreria opencv para abrir y manipular la imagen
2 import cv2
3
4 # Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
5 imagen = cv2.imread("rgb.jpg")
6
7 # Se genera una imagen en escala de grises
8 grises = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
9
10 # Se crea la imagen negativa de la imagen en escala de grises
11 negativa = 255 - grises
12
13 # Se muestra la imagen original con opencv
14 cv2.imshow("Imagen", imagen)
15 # Se muestra la imagen en escala de grises con opencv
16 cv2.imshow("Grises", grises)
17 # Se muestra el negativo de la imagen en escala de grises con opencv
18 cv2.imshow("Negativa", negativa)
19
20 # opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Imagen original



Escala de grises de la imagen

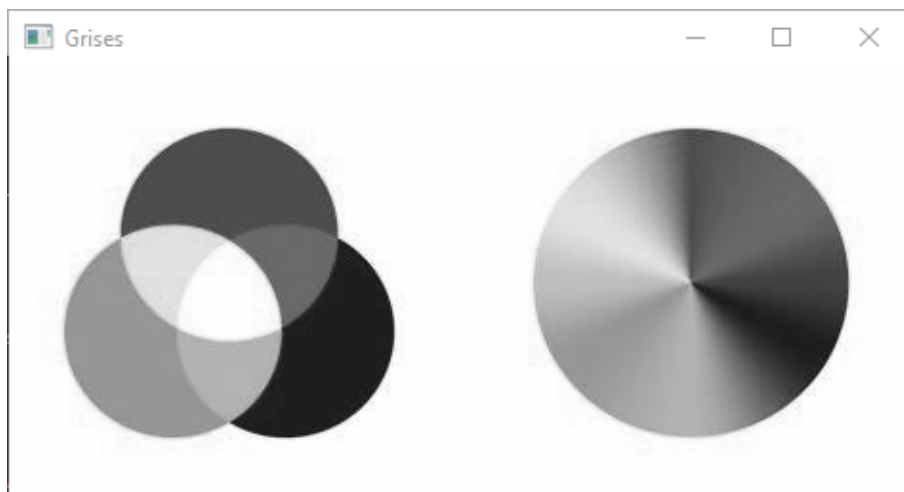
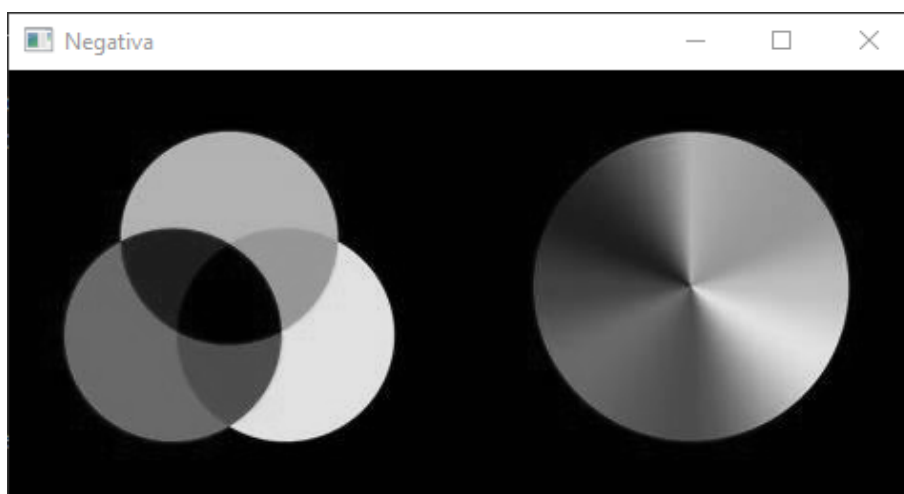


Imagen en negativo



Ejercicio 2

```
# Se importa la libreria opencv para abrir y manipular la imagen
import cv2

# Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
imagen = cv2.imread("rgb.jpg")

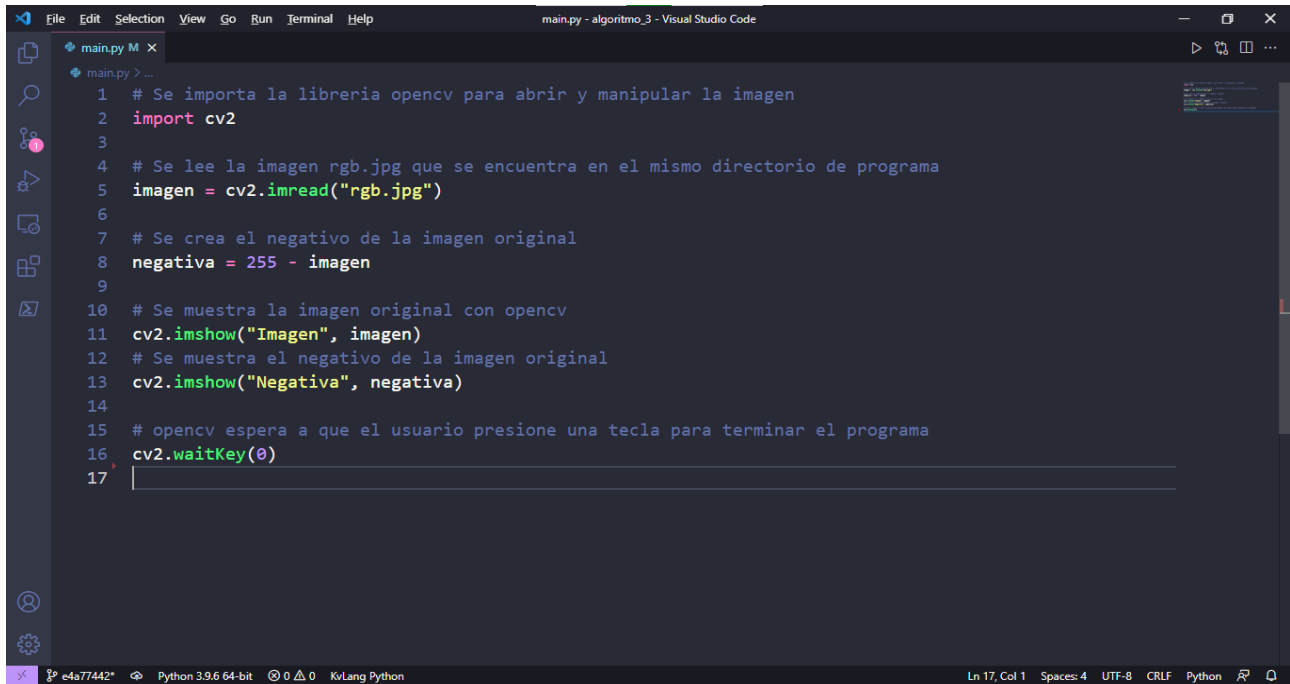
# Se crea el negativo de la imagen original
negativa = 255 - imagen

# Se muestra la imagen original con opencv
cv2.imshow("Imagen", imagen)
# Se muestra el negativo de la imagen original
cv2.imshow("Negativa", negativa)

# opencv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el ejercicio 2, primer se importa la librería opencv, después se carga la imagen "rgb.jpg" que se encuentra en el mismo directorio del programa, luego se crea el negativo de la imagen original. Se muestran las imágenes original y la negativa, al final opencv espera a que el usuario oprima una tecla para terminar el programa.

Imagen del código del ejercicio 2



The image shows a screenshot of the Visual Studio Code editor interface. The main editor window displays a Python script named `main.py` with the following code:

```
1 # Se importa la libreria opencv para abrir y manipular la imagen
2 import cv2
3
4 # Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
5 imagen = cv2.imread("rgb.jpg")
6
7 # Se crea el negativo de la imagen original
8 negativa = 255 - imagen
9
10 # Se muestra la imagen original con opencv
11 cv2.imshow("Imagen", imagen)
12 # Se muestra el negativo de la imagen original
13 cv2.imshow("Negativa", negativa)
14
15 # opencv espera a que el usuario presione una tecla para terminar el programa
16 cv2.waitKey(0)
17
```

The status bar at the bottom indicates the file encoding is UTF-8, the line ending is CRLF, and the language is Python. The cursor is positioned at line 17, column 1.

Imagen original

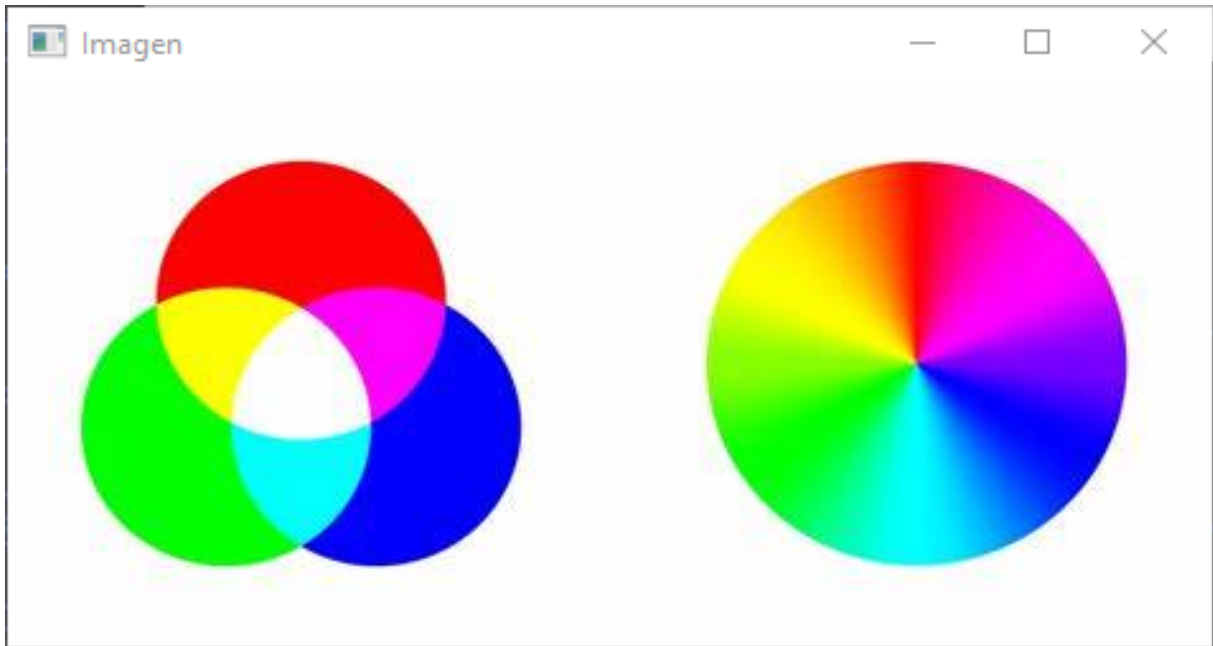
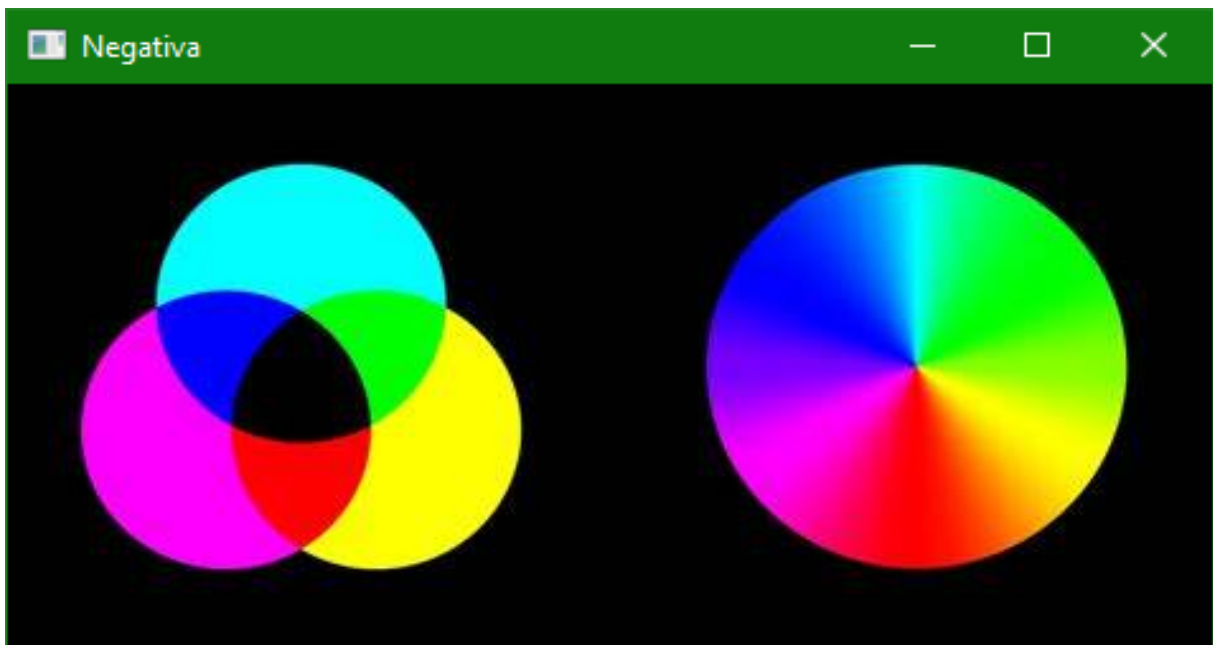


Imagen en negativo



Ejercicio 3

```
# Se importa la libreria opencv para abrir y manipular la imagen
import cv2

# Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
imagen = cv2.imread("rgb.jpg")

# Se gira la imagen original 90 grados en sentido de las manecillas y se asigna en la
# variable 'grises'
grises = cv2.rotate(imagen, cv2.ROTATE_90_CLOCKWISE)
# Se genera la imagen en escala de grises de la imagen rotada
grises = cv2.cvtColor(grises, cv2.COLOR_BGR2GRAY)

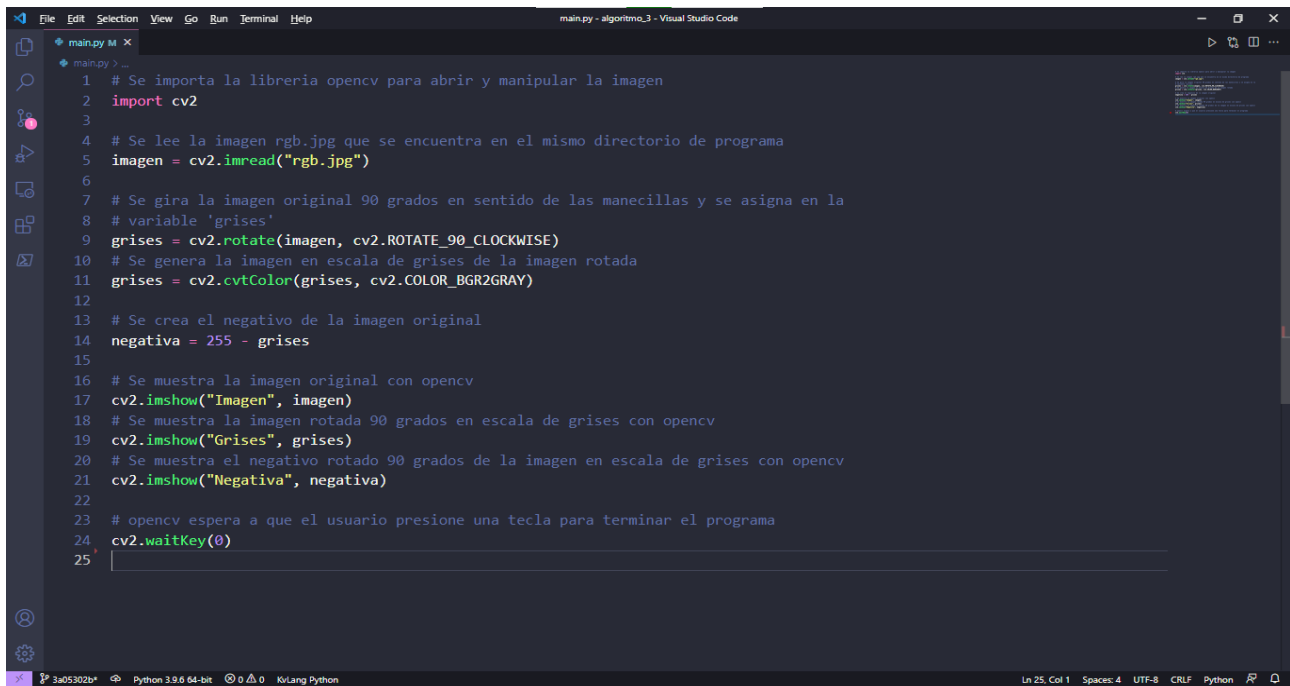
# Se crea el negativo de la imagen original
negativa = 255 - grises

# Se muestra la imagen original con opencv
cv2.imshow("Imagen", imagen)
# Se muestra la imagen rotada 90 grados en escala de grises con opencv
cv2.imshow("Grises", grises)
# Se muestra el negativo rotado 90 grados de la imagen en escala de grises con opencv
cv2.imshow("Negativa", negativa)

# opencv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el ejercicio 3, primero se importa la librería opencv, después se carga la imagen "rgb.jpg" que se encuentra en el mismo directorio del programa, luego se rota la imagen original 90° en el sentido de las manecillas del reloj y se cambia a escala de grises, luego se obtiene el negativo de la imagen rotada. Se muestran las imágenes original, la imagen rotada en escala de grises y la imagen negativa, al final opencv espera a que el usuario presione una tecla para terminar el programa.

Imagen del código del ejercicio 3



```
1 # Se importa la libreria opencv para abrir y manipular la imagen
2 import cv2
3
4 # Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
5 imagen = cv2.imread("rgb.jpg")
6
7 # Se gira la imagen original 90 grados en sentido de las manecillas y se asigna en la
8 # variable 'grises'
9 grises = cv2.rotate(imagen, cv2.ROTATE_90_CLOCKWISE)
10 # Se genera la imagen en escala de grises de la imagen rotada
11 grises = cv2.cvtColor(grises, cv2.COLOR_BGR2GRAY)
12
13 # Se crea el negativo de la imagen original
14 negativa = 255 - grises
15
16 # Se muestra la imagen original con opencv
17 cv2.imshow("Imagen", imagen)
18 # Se muestra la imagen rotada 90 grados en escala de grises con opencv
19 cv2.imshow("Grises", grises)
20 # Se muestra el negativo rotado 90 grados de la imagen en escala de grises con opencv
21 cv2.imshow("Negativa", negativa)
22
23 # opencv espera a que el usuario presione una tecla para terminar el programa
24 cv2.waitKey(0)
25
```


Imagen original

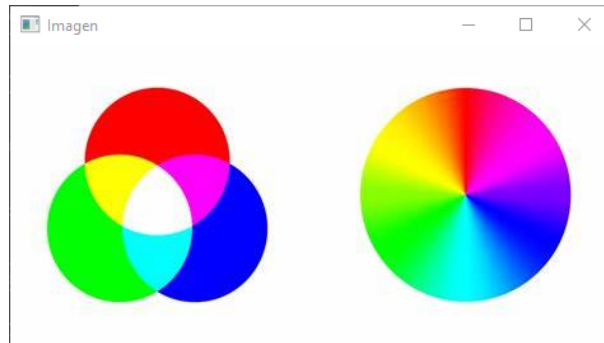


Imagen rotada en escala de grises

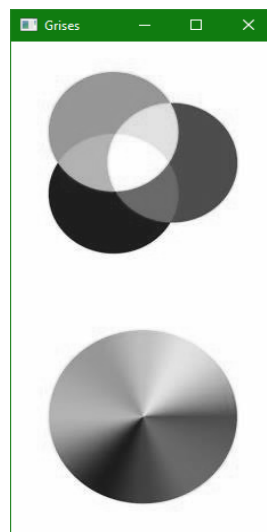
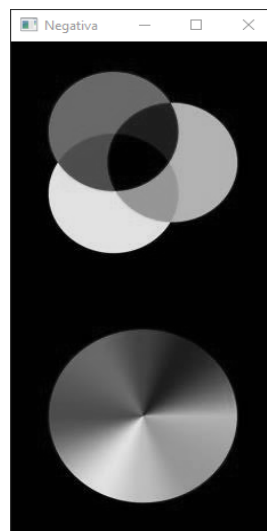


Imagen rotada en negativo



Ejercicio 4

```
# Se importa la libreria opencv para abrir y manipular la imagen
import cv2

# Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
imagen = cv2.imread("rgb.jpg")

# Se gira la imagen original 90 grados en sentido contrario de las manecillas y se asigna en la
# variable 'grises'
grises = cv2.rotate(imagen, cv2.ROTATE_90_COUNTERCLOCKWISE)
# Se genera la imagen en escala de grises de la imagen rotada
grises = cv2.cvtColor(grises, cv2.COLOR_BGR2GRAY)

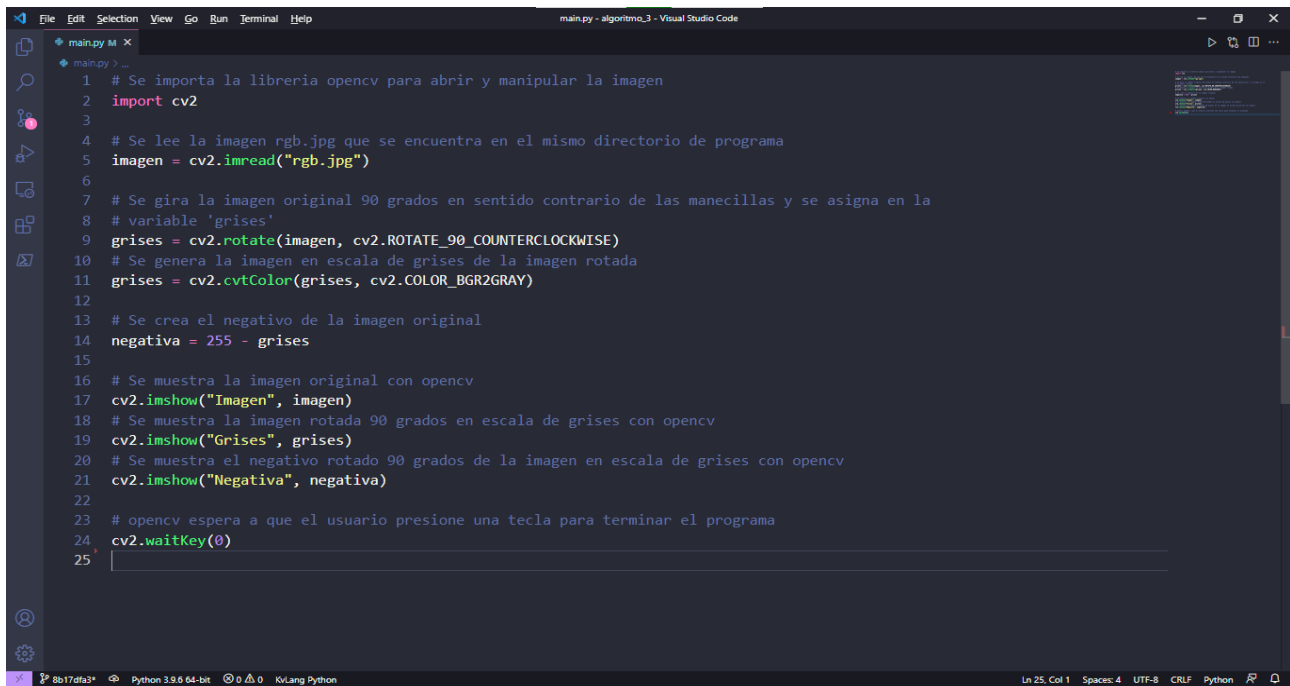
# Se crea el negativo de la imagen original
negativa = 255 - grises

# Se muestra la imagen original con opencv
cv2.imshow("Imagen", imagen)
# Se muestra la imagen rotada 90 grados en escala de grises con opencv
cv2.imshow("Grises", grises)
# Se muestra el negativo rotado 90 grados de la imagen en escala de grises con opencv
cv2.imshow("Negativa", negativa)

# opencv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el ejercicio 4, primero se importa la librería opencv, después se carga la imagen "rgb.jpg" que se encuentra en el mismo directorio del programa, luego se rota la imagen original 90° en sentido contrario de las manecillas del reloj, y se convierte a escala de grises, luego se obtiene la negativa de la imagen rotada en escala de grises. Por último se muestran las imágenes original, la imagen rotada en escala de grises y la rotada en negativa. Al final opencv espera a que el usuario presione una tecla para terminar el programa.

Imagen del código del ejercicio 4



```
1 # Se importa la libreria opencv para abrir y manipular la imagen
2 import cv2
3
4 # Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
5 imagen = cv2.imread("rgb.jpg")
6
7 # Se gira la imagen original 90 grados en sentido contrario de las manecillas y se asigna en la
8 # variable 'grises'
9 grises = cv2.rotate(imagen, cv2.ROTATE_90_COUNTERCLOCKWISE)
10 # Se genera la imagen en escala de grises de la imagen rotada
11 grises = cv2.cvtColor(grises, cv2.COLOR_BGR2GRAY)
12
13 # Se crea el negativo de la imagen original
14 negativa = 255 - grises
15
16 # Se muestra la imagen original con opencv
17 cv2.imshow("Imagen", imagen)
18 # Se muestra la imagen rotada 90 grados en escala de grises con opencv
19 cv2.imshow("Grises", grises)
20 # Se muestra el negativo rotado 90 grados de la imagen en escala de grises con opencv
21 cv2.imshow("Negativa", negativa)
22
23 # opencv espera a que el usuario presione una tecla para terminar el programa
24 cv2.waitKey(0)
25
```

Imagen original

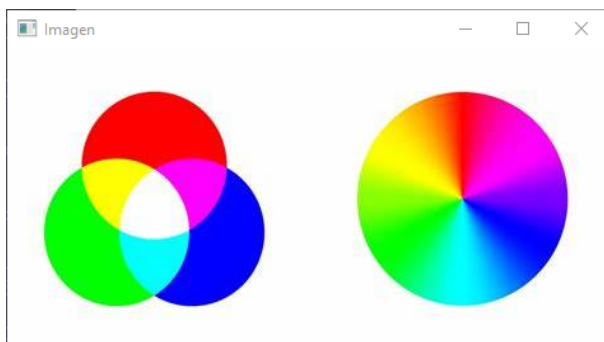


Imagen rotada en escala de grises

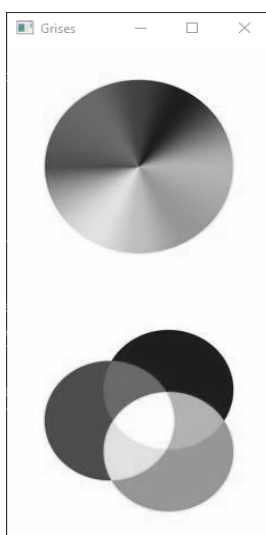
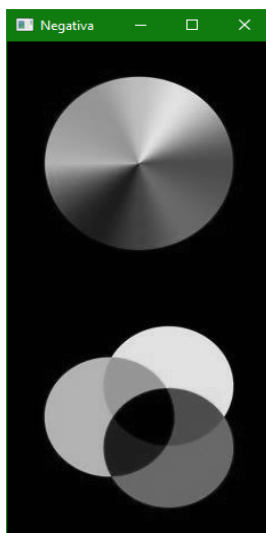


Imagen rotada en negativo



Ejercicio 5

```
# Se importa la libreria opencv para abrir y manipular la imagen
import cv2

# Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
imagen = cv2.imread("rgb.jpg")

# Se gira la imagen original 90 grados en sentido contrario de las manecillas y se asigna en la
# variable 'grises'
grises = cv2.rotate(imagen, cv2.ROTATE_180)
# Se genera la imagen en escala de grises de la imagen rotada
grises = cv2.cvtColor(grises, cv2.COLOR_BGR2GRAY)

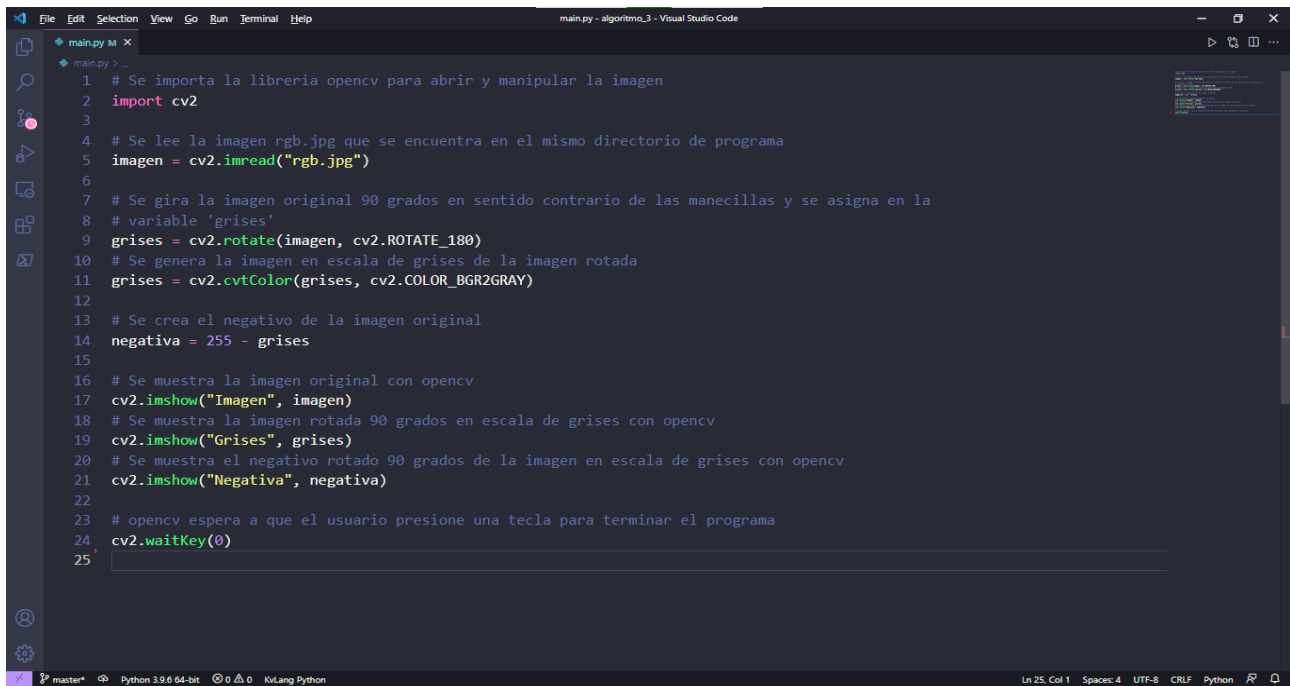
# Se crea el negativo de la imagen original
negativa = 255 - grises

# Se muestra la imagen original con opencv
cv2.imshow("Imagen", imagen)
# Se muestra la imagen rotada 90 grados en escala de grises con opencv
cv2.imshow("Grises", grises)
# Se muestra el negativo rotado 90 grados de la imagen en escala de grises con opencv
cv2.imshow("Negativa", negativa)

# opencv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el ejercicio 5, primero se importa la librería opencv, después se carga la imagen "rgb.jpg" que se encuentra en el mismo directorio del programa, luego se rota la imagen 180° y se convierte a escala de grises, luego se obtiene la negativa de la imagen rotada en escala de grises. Por último se muestran las imágenes original, en escala de grises y negativa. Al final opencv espera a que el usuario oprima una tecla para terminar el programa.

Imagen del código del ejercicio 5



```
1 # Se importa la libreria opencv para abrir y manipular la imagen
2 import cv2
3
4 # Se lee la imagen rgb.jpg que se encuentra en el mismo directorio de programa
5 imagen = cv2.imread("rgb.jpg")
6
7 # Se gira la imagen original 90 grados en sentido contrario de las manecillas y se asigna en la
8 # variable 'grises'
9 grises = cv2.rotate(imagen, cv2.ROTATE_180)
10 # Se genera la imagen en escala de grises de la imagen rotada
11 grises = cv2.cvtColor(grises, cv2.COLOR_BGR2GRAY)
12
13 # Se crea el negativo de la imagen original
14 negativa = 255 - grises
15
16 # Se muestra la imagen original con opencv
17 cv2.imshow("Imagen", imagen)
18 # Se muestra la imagen rotada 90 grados en escala de grises con opencv
19 cv2.imshow("Grises", grises)
20 # Se muestra el negativo rotado 90 grados de la imagen en escala de grises con opencv
21 cv2.imshow("Negativa", negativa)
22
23 # opencv espera a que el usuario presione una tecla para terminar el programa
24 cv2.waitKey(0)
25
```

Imagen original

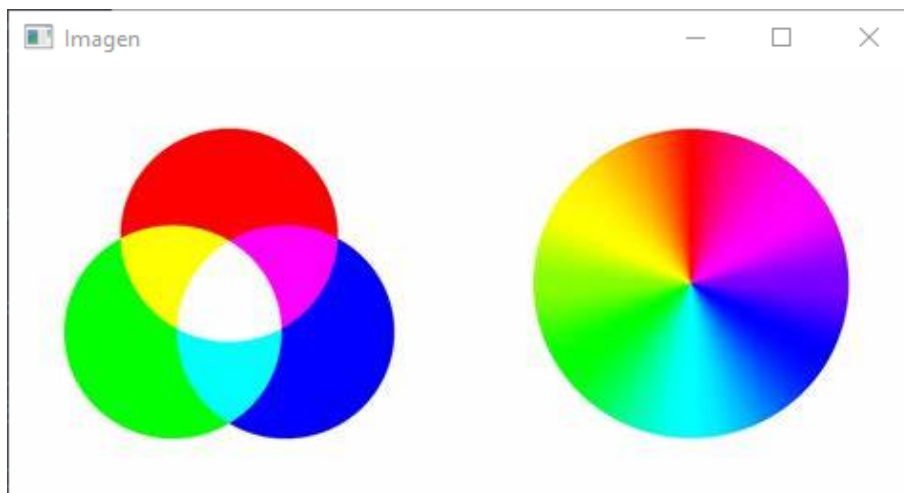


Imagen rotada en escala de grises

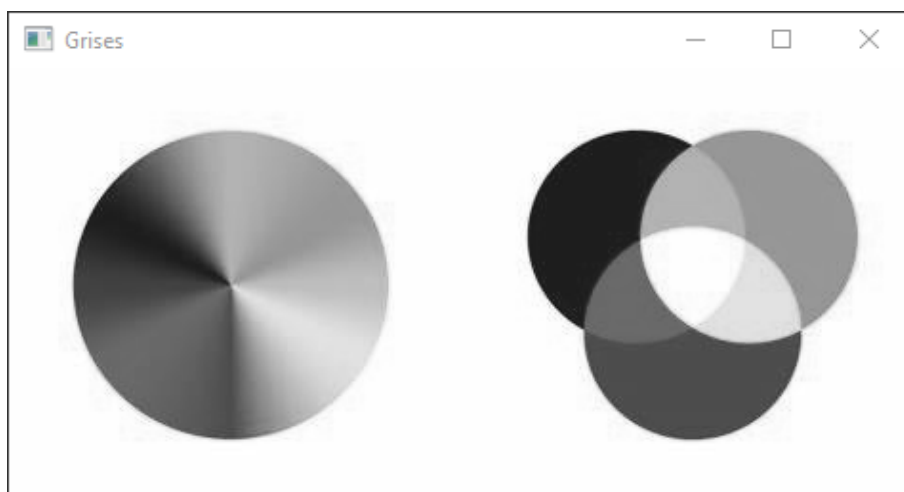


Imagen rotada en negativo

