

Algoritmo 8

Muñoz Nuñez Ian Emmanuel

Visión Robótica

"Operaciones Mascaras Imágenes"

Ejercicio 1

```
# Se importa opencv
import cv2

# Se lee la imagen 'sonic.jpg'
imagen = cv2.imread("sonic.jpg")

# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)

# Se cambia el formato de la imagen de BGR a HSV
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

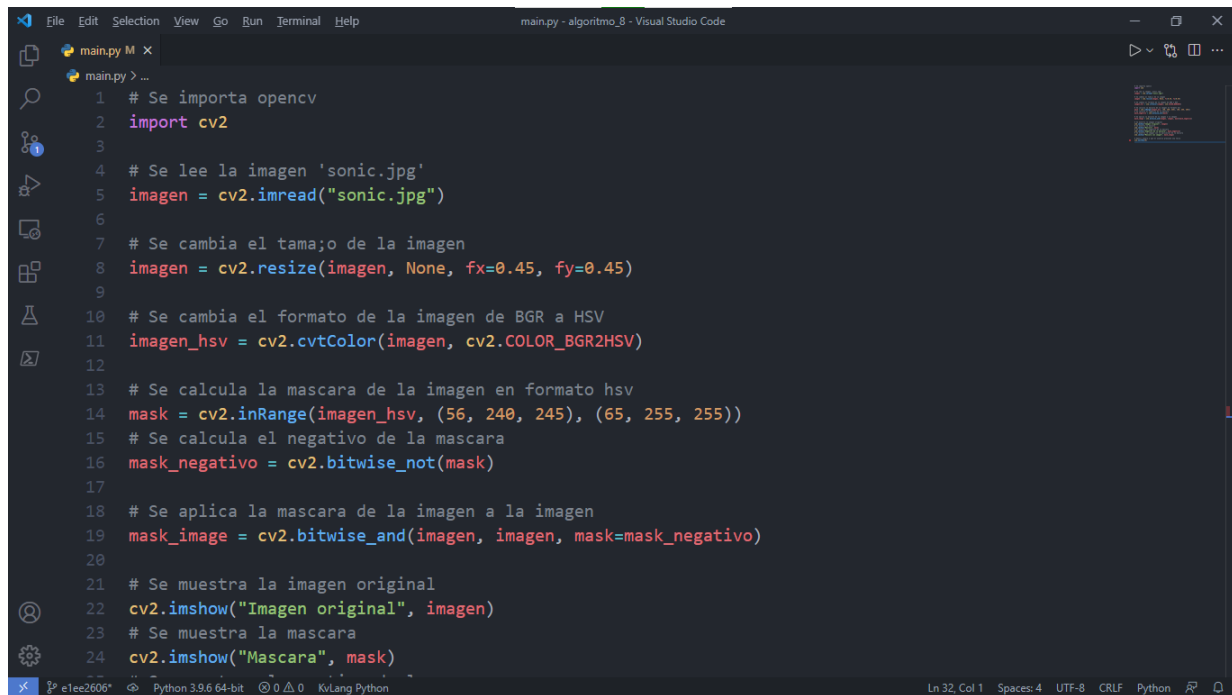
# Se calcula la mascara de la imagen en formato hsv
mask = cv2.inRange(imagen_hsv, (56, 240, 245), (65, 255, 255))
# Se calcula el negativo de la mascara
mask_negativo = cv2.bitwise_not(mask)

# Se aplica la mascara de la imagen a la imagen
mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)

# Se muestra la imagen original
cv2.imshow("Imagen original", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el negativo de la mascara
cv2.imshow("Negativo de la mascara", mask_negativo)
# Se muestra la imagen con la aplicacion de la mascara
cv2.imshow("Mascara con imagen", mask_image)

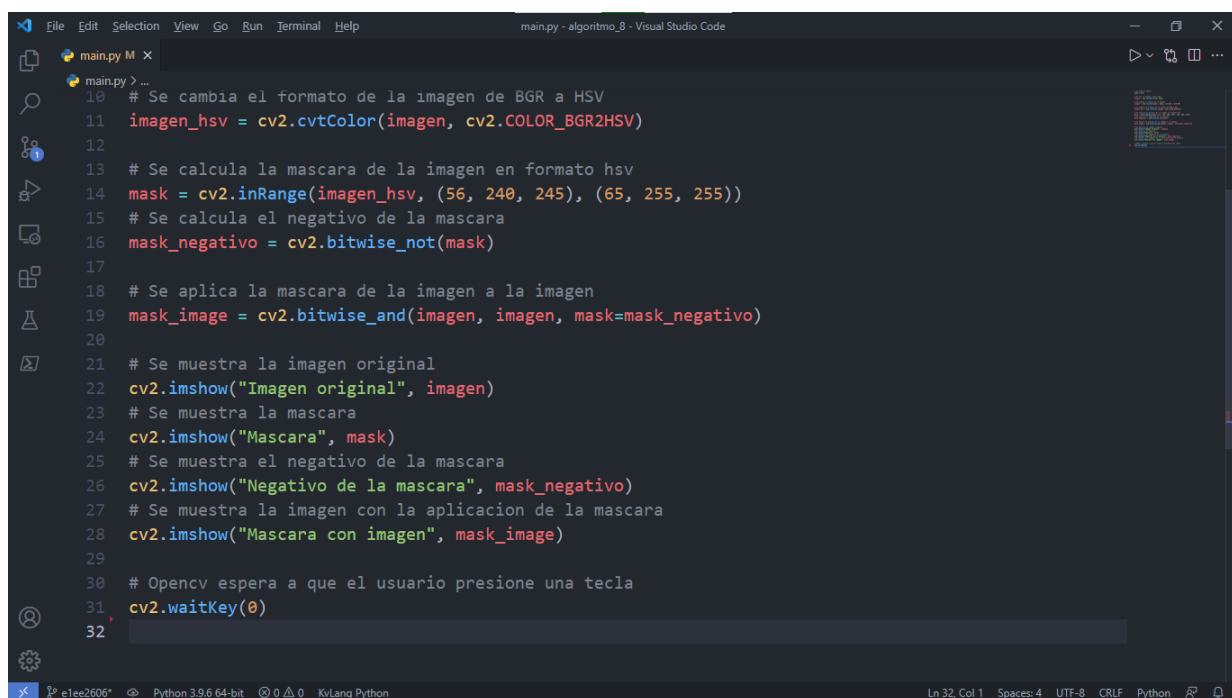
# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 1 primero se importa **opencv**, se lee la imagen "sonic.jpg" y se modifica su tamaño. Luego se cambia su formato de BGR a HSV, después se calcula la mascara de la imagen y su negativo. Por último se aplica la mascara a la imagen y se muestran la imagen original, su mascara, el negativo de la mascara y la imagen con la aplicación de la mascara. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.



```
1 # Se importa opencv
2 import cv2
3
4 # Se lee la imagen 'sonic.jpg'
5 imagen = cv2.imread("sonic.jpg")
6
7 # Se cambia el tamaño de la imagen
8 imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)
9
10 # Se cambia el formato de la imagen de BGR a HSV
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (56, 240, 245), (65, 255, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
```

Figura 1: Captura 1 del código del ejercicio 1



```
10 # Se cambia el formato de la imagen de BGR a HSV
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (56, 240, 245), (65, 255, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
25 # Se muestra el negativo de la mascara
26 cv2.imshow("Negativo de la mascara", mask_negativo)
27 # Se muestra la imagen con la aplicacion de la mascara
28 cv2.imshow("Mascara con imagen", mask_image)
29
30 # Opencv espera a que el usuario presione una tecla
31 cv2.waitKey(0)
32
```

Figura 2: Captura 2 del código del ejercicio 1

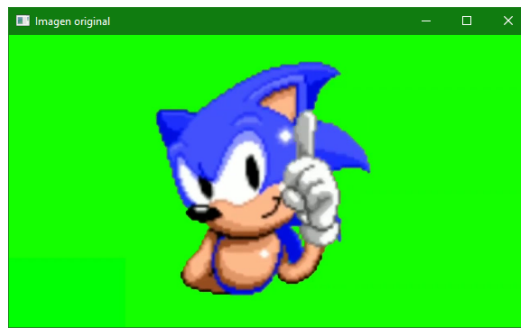


Figura 3: Imagen original



Figura 4: Mascara de la imagen original



Figura 5: Negativo de la mascara

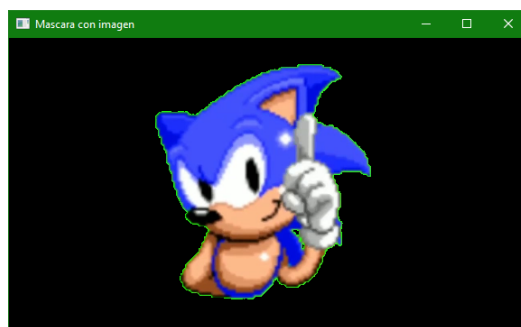


Figura 6: Aplicación de la mascara a la imagen

Ejercicio 2 "Imagen: Batman"

```
# Se importa opencv
import cv2

# Se lee la imagen 'batman.jpg'
imagen = cv2.imread("batman.jpg")

# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)

# Se cambia el formato de la imagen de BGR a HSV
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

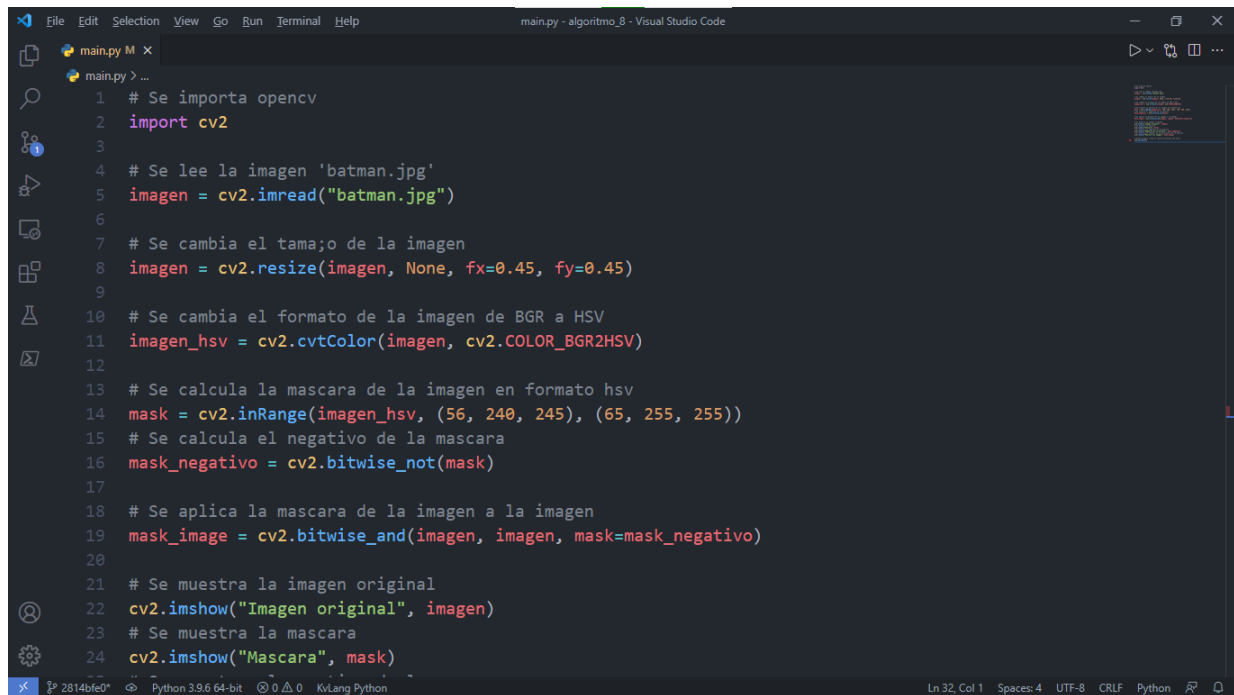
# Se calcula la mascara de la imagen en formato hsv
mask = cv2.inRange(imagen_hsv, (56, 240, 245), (65, 255, 255))
# Se calcula el negativo de la mascara
mask_negativo = cv2.bitwise_not(mask)

# Se aplica la mascara de la imagen a la imagen
mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)

# Se muestra la imagen original
cv2.imshow("Imagen original", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el negativo de la mascara
cv2.imshow("Negativo de la mascara", mask_negativo)
# Se muestra la imagen con la aplicacion de la mascara
cv2.imshow("Mascara con imagen", mask_image)

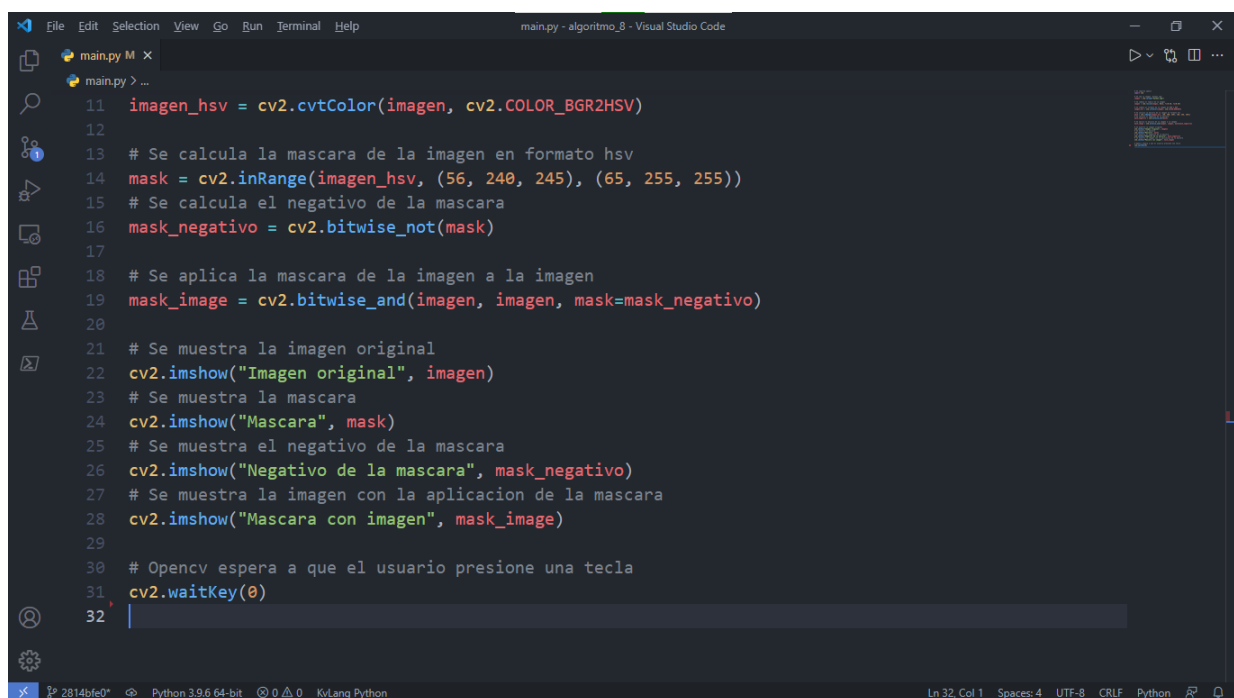
# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 2 con la imagen "Batman" primero se importa **opencv**, se lee la imagen "batman.jpg" y se modifica su tamaño. Luego se cambia su formato de BGR a HSV, después se calcula la mascara de la imagen y su negativo. Por último se aplica la mascara a la imagen y se muestran la imagen original, su mascara, el negativo de la mascara y la imagen con la aplicación de la mascara. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.



```
1 # Se importa opencv
2 import cv2
3
4 # Se lee la imagen 'batman.jpg'
5 imagen = cv2.imread("batman.jpg")
6
7 # Se cambia el tamaño de la imagen
8 imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)
9
10 # Se cambia el formato de la imagen de BGR a HSV
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (56, 240, 245), (65, 255, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
```

Figura 7: Captura 1 del código del ejercicio 2 para la imagen "batman.jpg"



```
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (56, 240, 245), (65, 255, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
25 # Se muestra el negativo de la mascara
26 cv2.imshow("Negativo de la mascara", mask_negativo)
27 # Se muestra la imagen con la aplicacion de la mascara
28 cv2.imshow("Mascara con imagen", mask_image)
29
30 # Opencv espera a que el usuario presione una tecla
31 cv2.waitKey(0)
32
```

Figura 8: Captura 2 del código del ejercicio 2 para la imagen "batman.jpg"

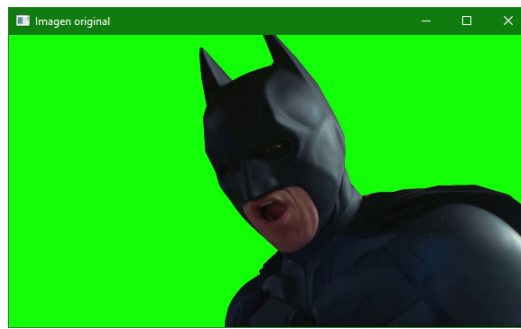


Figura 9: Imagen original del ejercicio 2 con la imagen "batman.jpg"



Figura 10: Mascara de la imagen

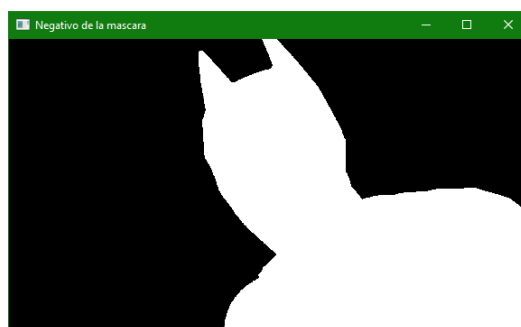


Figura 11: Negativo de la mascara

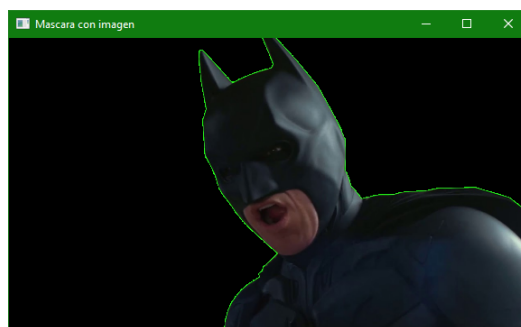


Figura 12: Aplicación de la mascara a la imagen

Ejercicio 2 "Imagen: Capitán América"

```
# Se importa opencv
import cv2

# Se lee la imagen 'capitan.jpg'
imagen = cv2.imread("capitan.jpg")

# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)

# Se cambia el formato de la imagen de BGR a HSV
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

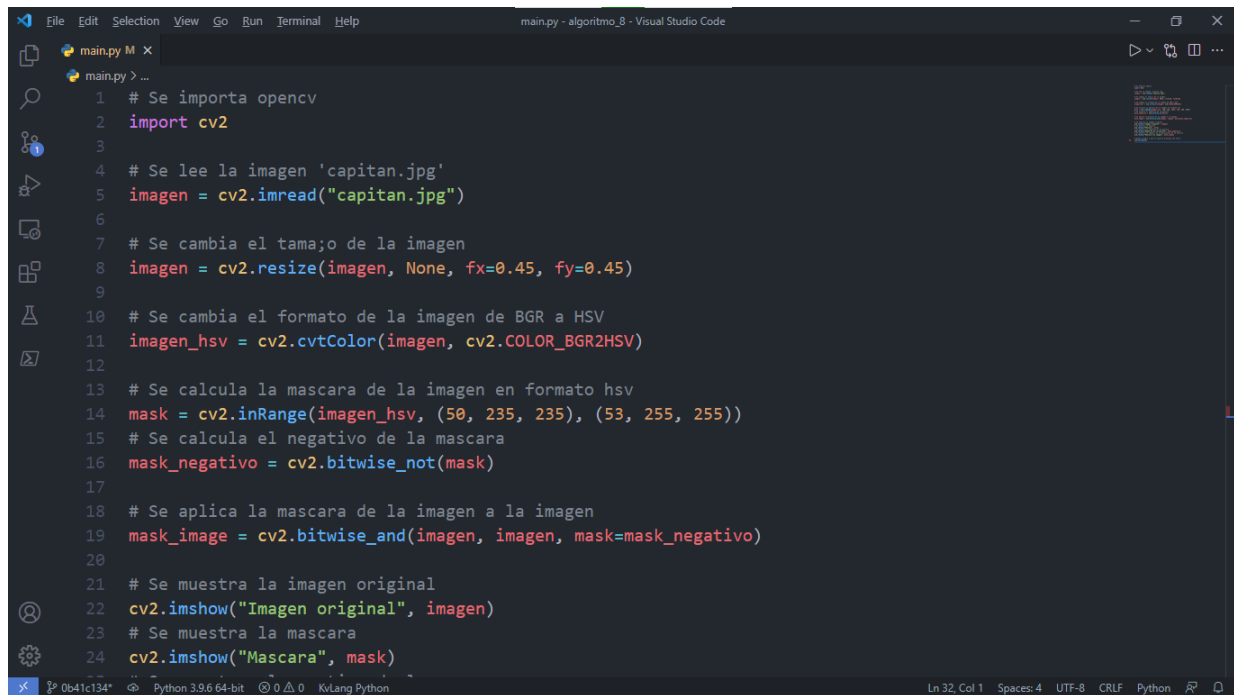
# Se calcula la mascara de la imagen en formato hsv
mask = cv2.inRange(imagen_hsv, (50, 235, 235), (53, 255, 255))
# Se calcula el negativo de la mascara
mask_negativo = cv2.bitwise_not(mask)

# Se aplica la mascara de la imagen a la imagen
mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)

# Se muestra la imagen original
cv2.imshow("Imagen original", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el negativo de la mascara
cv2.imshow("Negativo de la mascara", mask_negativo)
# Se muestra la imagen con la aplicacion de la mascara
cv2.imshow("Mascara con imagen", mask_image)

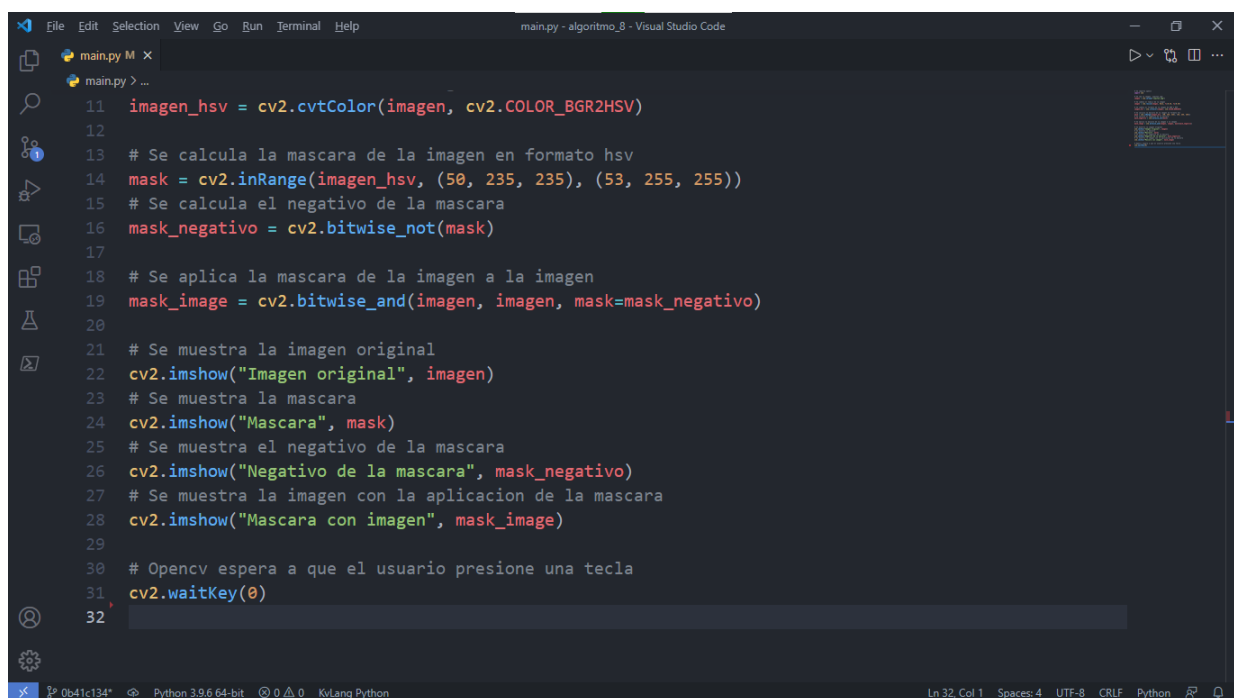
# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 2 con la imagen "Capitán" primero se importa **opencv**, se lee la imagen "capitan.jpg" y se modifica su tamaño. Luego se cambia su formato de BGR a HSV, después se calcula la mascara de la imagen y su negativo. Por último se aplica la mascara a la imagen y se muestran la imagen original, su mascara, el negativo de la mascara y la imagen con la aplicación de la mascara. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.



```
1 # Se importa opencv
2 import cv2
3
4 # Se lee la imagen 'capitan.jpg'
5 imagen = cv2.imread("capitan.jpg")
6
7 # Se cambia el tamaño de la imagen
8 imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)
9
10 # Se cambia el formato de la imagen de BGR a HSV
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (50, 235, 235), (53, 255, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
```

Figura 13: Captura 1 del código del ejercicio 2 para la imagen "capitan.jpg"



```
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (50, 235, 235), (53, 255, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
25 # Se muestra el negativo de la mascara
26 cv2.imshow("Negativo de la mascara", mask_negativo)
27 # Se muestra la imagen con la aplicacion de la mascara
28 cv2.imshow("Mascara con imagen", mask_image)
29
30 # Opencv espera a que el usuario presione una tecla
31 cv2.waitKey(0)
32
```

Figura 14: Captura 2 del código del ejercicio 2 para la imagen "capitan.jpg"

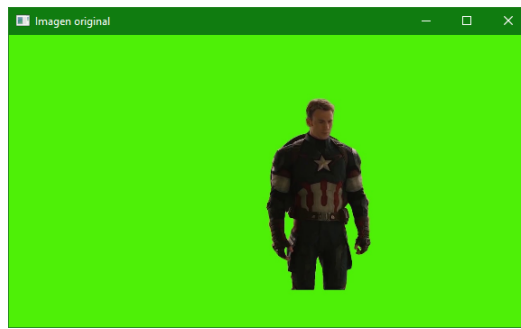


Figura 15: Imagen original del ejercicio 2 con la imagen "capitan.jpg"

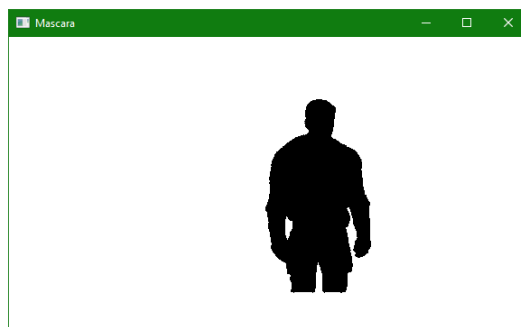


Figura 16: Mascara de la imagen



Figura 17: Negativo de la mascara

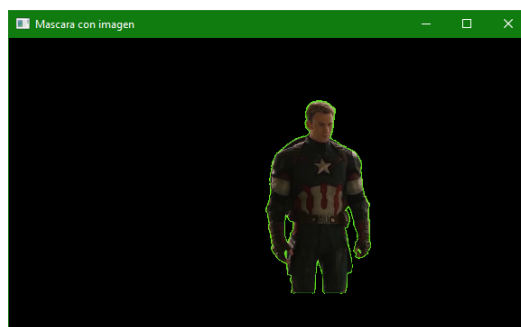


Figura 18: Aplicación de la mascara a la imagen

Ejercicio 2 "Iron-Man"

```
# Se importa opencv
import cv2

# Se lee la imagen 'ironman.jpg'
imagen = cv2.imread("ironman.jpg")

# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)

# Se cambia el formato de la imagen de BGR a HSV
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

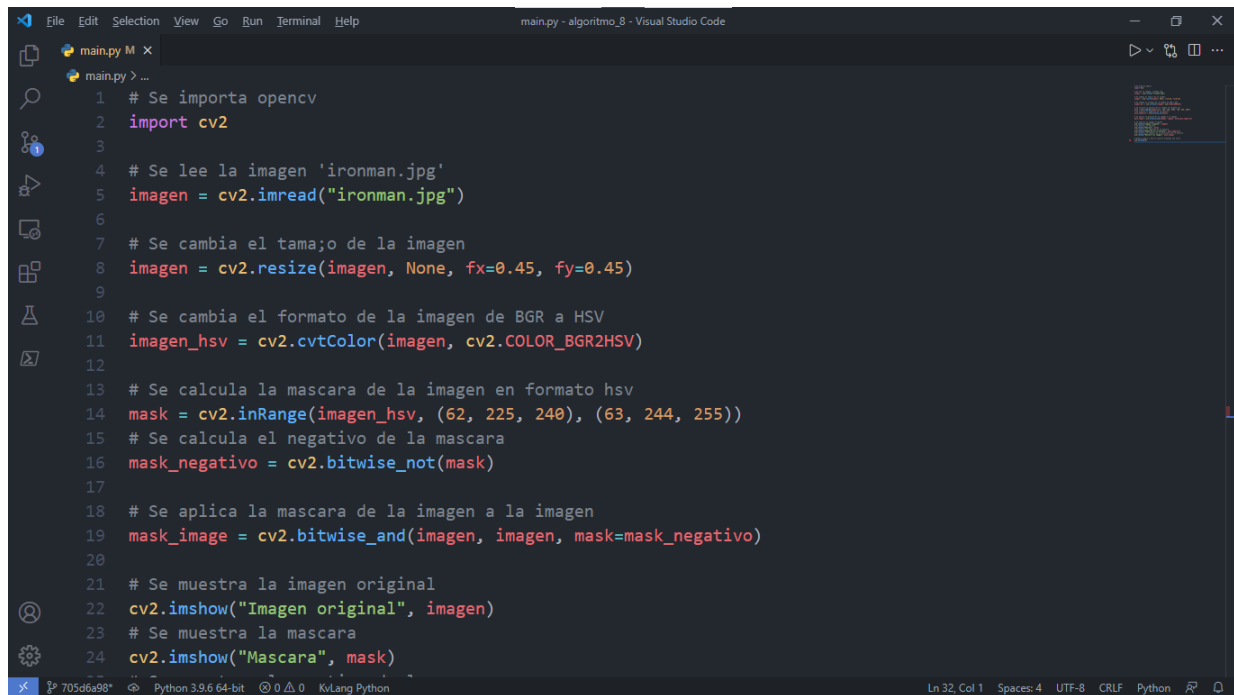
# Se calcula la mascara de la imagen en formato hsv
mask = cv2.inRange(imagen_hsv, (62, 225, 240), (63, 244, 255))
# Se calcula el negativo de la mascara
mask_negativo = cv2.bitwise_not(mask)

# Se aplica la mascara de la imagen a la imagen
mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)

# Se muestra la imagen original
cv2.imshow("Imagen original", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el negativo de la mascara
cv2.imshow("Negativo de la mascara", mask_negativo)
# Se muestra la imagen con la aplicacion de la mascara
cv2.imshow("Mascara con imagen", mask_image)

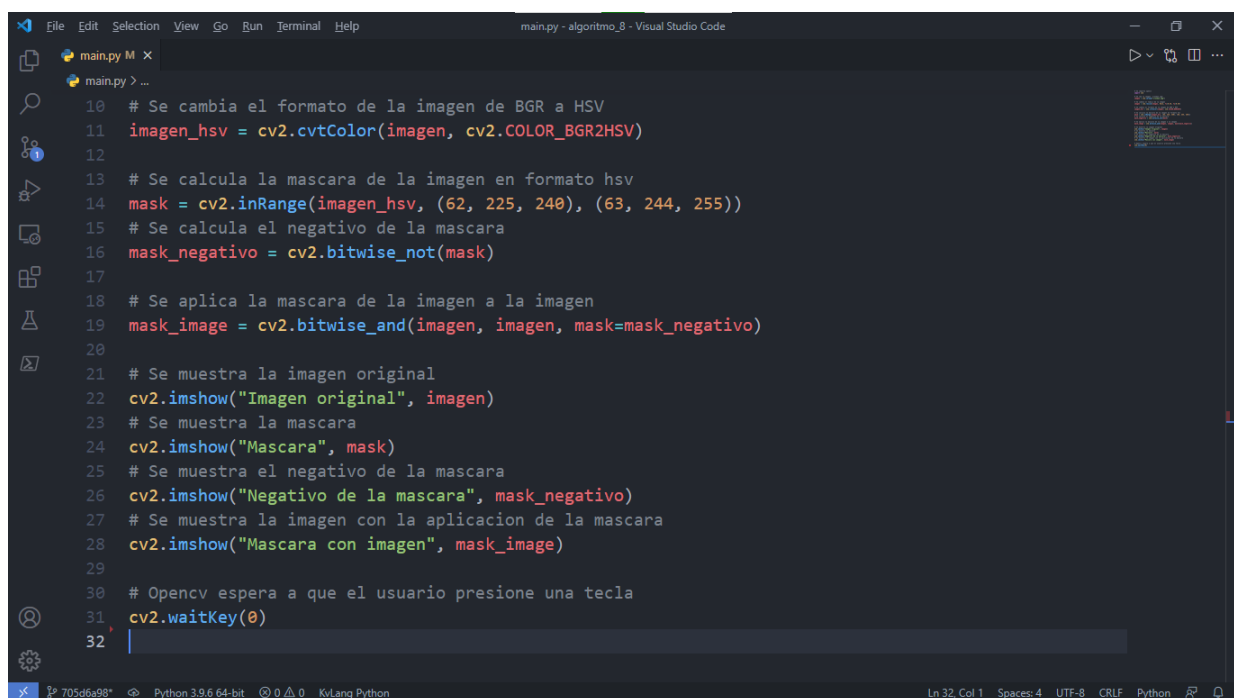
# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 2 con la imagen "Iron-Man" primero se importa **opencv**, se lee la imagen "ironman.jpg" y se modifica su tamaño. Luego se cambia su formato de BGR a HSV, después se calcula la mascara de la imagen y su negativo. Por último se aplica la mascara a la imagen y se muestran la imagen original, su mascara, el negativo de la mascara y la imagen con la aplicación de la mascara. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.



```
1 # Se importa opencv
2 import cv2
3
4 # Se lee la imagen 'ironman.jpg'
5 imagen = cv2.imread("ironman.jpg")
6
7 # Se cambia el tamaño de la imagen
8 imagen = cv2.resize(imagen, None, fx=0.45, fy=0.45)
9
10 # Se cambia el formato de la imagen de BGR a HSV
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (62, 225, 240), (63, 244, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
```

Figura 19: Captura 1 del código del ejercicio 2 para la imagen "ironman.jpg"



```
10 # Se cambia el formato de la imagen de BGR a HSV
11 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
12
13 # Se calcula la mascara de la imagen en formato hsv
14 mask = cv2.inRange(imagen_hsv, (62, 225, 240), (63, 244, 255))
15 # Se calcula el negativo de la mascara
16 mask_negativo = cv2.bitwise_not(mask)
17
18 # Se aplica la mascara de la imagen a la imagen
19 mask_image = cv2.bitwise_and(imagen, imagen, mask=mask_negativo)
20
21 # Se muestra la imagen original
22 cv2.imshow("Imagen original", imagen)
23 # Se muestra la mascara
24 cv2.imshow("Mascara", mask)
25 # Se muestra el negativo de la mascara
26 cv2.imshow("Negativo de la mascara", mask_negativo)
27 # Se muestra la imagen con la aplicacion de la mascara
28 cv2.imshow("Mascara con imagen", mask_image)
29
30 # Opencv espera a que el usuario presione una tecla
31 cv2.waitKey(0)
32
```

Figura 20: Captura 2 del código del ejercicio 2 para la imagen "ironman.jpg"

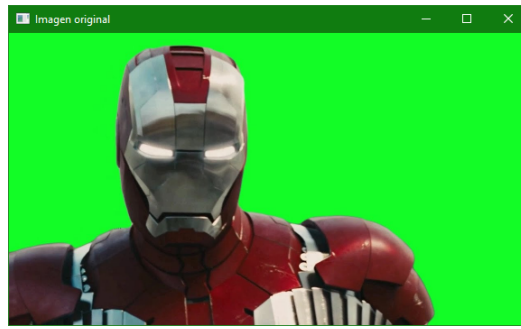


Figura 21: Imagen original del ejercicio 2 con la imagen "ironman.jpg"

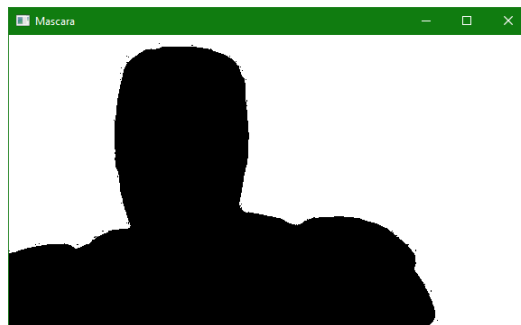


Figura 22: Mascara de la imagen



Figura 23: Negativo de la imagen

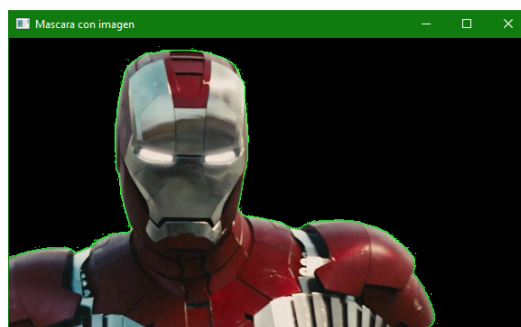


Figura 24: Aplicación de la mascara a la imagen