

Algoritmo 7

Muñoz Nuñez Ian Emmanuel

Máscaras de imagen

Visión Robótica

```
# Se importa la libreria opencv
import cv2

# Se lee la imagen 'ejercicio1.png'
imagen = cv2.imread("ejercicio1.png")
# Se cambia el tamaño de la imagen para ser vista de mejor manera
imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
# Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva
variable
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se obtiene la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (150, 120, 120), (150, 170, 170))

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara de la imagen
cv2.imshow("Mascara", mask)
# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 1 primero se importa la librería **opencv**. Luego se lee la imagen "ejercicio1.png" que se encuentra en el mismo directorio del algoritmo, después se modifica su tamaño para poder ver la imagen de mejor manera y se cambia su formato de BGR a HSV. Por último se calcula la mascara de la imagen y se muestra la imagen original y la mascara de esta. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.

```
1 # Se importa la libreria opencv
2 import cv2
3
4 # Se lee la imagen 'ejercicio1.png'
5 imagen = cv2.imread("ejercicio1.png")
6 # Se cambia el tamaño de la imagen para ser vista de mejor manera
7 imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
8 # Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva variable
9 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
10 # Se obtiene la mascara de la imagen
11 mask = cv2.inRange(imagen_hsv, (150, 120, 120), (150, 170, 170))
12
13 # Se muestra la imagen original
14 cv2.imshow("Imagen", imagen)
15 # Se muestra la mascara de la imagen
16 cv2.imshow("Mascara", mask)
17 # Opencv espera a que el usuario presione una tecla
18 cv2.waitKey(0)
19
```

Figura 1: Imagen del código del ejercicio 1

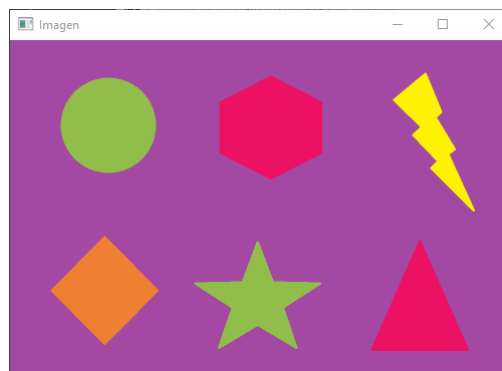


Figura 2: Imagen original del ejercicio 1

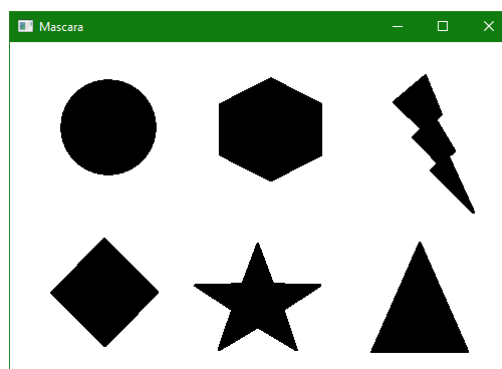


Figura 3: Mascara de imagen del ejercicio 1

```
# Se importa la libreria opencv
import cv2

# Se lee la imagen 'ejercicio2.png'
imagen = cv2.imread("ejercicio2.png")
# Se cambia el tamaño de la imagen para ser vista de mejor manera
imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
# Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva
variable
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se obtiene la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (27, 175, 175), (27, 190, 190))

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara de la imagen
cv2.imshow("Mascara", mask)
# Opcv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 2 primero se importa la librería **opencv**. Luego se lee la imagen "ejercicio1.png" que se encuentra en el mismo directorio del algoritmo, después se modifica su tamaño para poder ver la imagen de mejor manera y se cambia su formato de BGR a HSV. Por último se calcula la mascara de la imagen y se muestra la imagen original y la mascara de esta. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.

```
1 # Se importa la libreria opencv
2 import cv2
3
4 # Se lee la imagen 'ejercicio2.png'
5 imagen = cv2.imread("ejercicio2.png")
6 # Se cambia el tamaño de la imagen para ser vista de mejor manera
7 imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
8 # Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva variable
9 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
10 # Se obtiene la mascara de la imagen
11 mask = cv2.inRange(imagen_hsv, (27, 175, 175), (27, 190, 190))
12
13 # Se muestra la imagen original
14 cv2.imshow("Imagen", imagen)
15 # Se muestra la mascara de la imagen
16 cv2.imshow("Mascara", mask)
17 # Opencv espera a que el usuario presione una tecla
18 cv2.waitKey(0)
19
```

Figura 4: Imagen del código del ejercicio 2



Figura 5: Imagen original del ejercicio 2

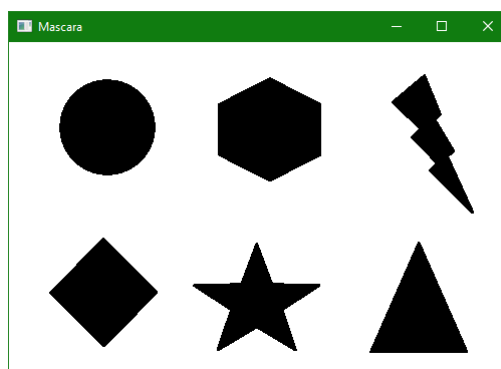


Figura 6: Mascara de imagen del ejercicio 2

```
# Se importa la libreria opencv
import cv2

# Se lee la imagen 'ejercicio3.png'
imagen = cv2.imread("ejercicio3.png")
# Se cambia el tamaño de la imagen para ser vista de mejor manera
imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
# Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva
variable
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se obtiene la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (89, 80, 80), (89, 150, 150))

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara de la imagen
cv2.imshow("Mascara", mask)
# Opcv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 3 primero se importa la librería **opencv**. Luego se lee la imagen "ejercicio1.png" que se encuentra en el mismo directorio del algoritmo, después se modifica su tamaño para poder ver la imagen de mejor manera y se cambia su formato de BGR a HSV. Por último se calcula la mascara de la imagen y se muestra la imagen original y la mascara de esta. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.

```
1 # Se importa la libreria opencv
2 import cv2
3
4 # Se lee la imagen 'ejercicio3.png'
5 imagen = cv2.imread("ejercicio3.png")
6 # Se cambia el tamaño de la imagen para ser vista de mejor manera
7 imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
8 # Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva variable
9 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
10 # Se obtiene la mascara de la imagen
11 mask = cv2.inRange(imagen_hsv, (89, 80, 80), (89, 150, 150))
12
13 # Se muestra la imagen original
14 cv2.imshow("Imagen", imagen)
15 # Se muestra la mascara de la imagen
16 cv2.imshow("Mascara", mask)
17 # Opencv espera a que el usuario presione una tecla
18 cv2.waitKey(0)
19
```

Figura 7: Imagen del código del ejercicio 3



Figura 8: Imagen original del ejercicio 3

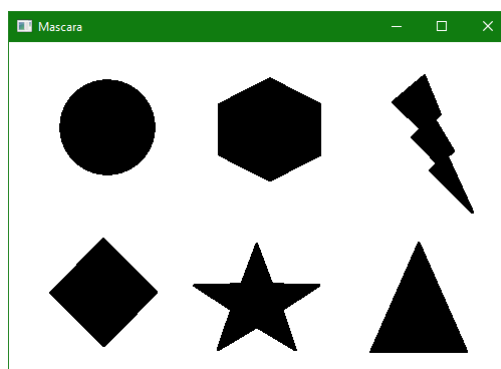


Figura 9: Mascara de imagen del ejercicio 3

```
# Se importa la libreria opencv
import cv2

# Se lee la imagen 'ejercicio4.png'
imagen = cv2.imread("ejercicio4.png")
# Se cambia el tamaño de la imagen para ser vista de mejor manera
imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
# Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva
variable
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se obtiene la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (160, 147, 167), (160, 147, 167))

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara de la imagen
cv2.imshow("Mascara", mask)
# Opcv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 4 primero se importa la librería ***opencv***. Luego se lee la imagen "ejercicio1.png" que se encuentra en el mismo directorio del algoritmo, después se modifica su tamaño para poder ver la imagen de mejor manera y se cambia su formato de BGR a HSV. Por último se calcula la mascara de la imagen y se muestra la imagen original y la mascara de esta. Al final ***opencv*** espera a que el usuario presione una tecla para terminar el programa.

```
1 # Se importa la libreria opencv
2 import cv2
3
4 # Se lee la imagen 'ejercicio4.png'
5 imagen = cv2.imread("ejercicio4.png")
6 # Se cambia el tamaño de la imagen para ser vista de mejor manera
7 imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
8 # Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva variable
9 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
10 # Se obtiene la mascara de la imagen
11 mask = cv2.inRange(imagen_hsv, (160, 147, 167), (160, 147, 167))
12
13 # Se muestra la imagen original
14 cv2.imshow("Imagen", imagen)
15 # Se muestra la mascara de la imagen
16 cv2.imshow("Mascara", mask)
17 # Opencv espera a que el usuario presione una tecla
18 cv2.waitKey(0)
19
```

Figura 10: Imagen del código del ejercicio 4

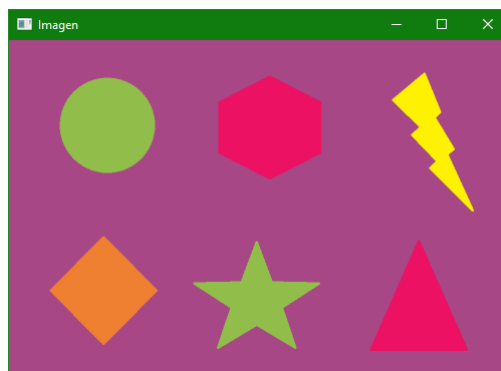


Figura 11: Imagen original del ejercicio 4

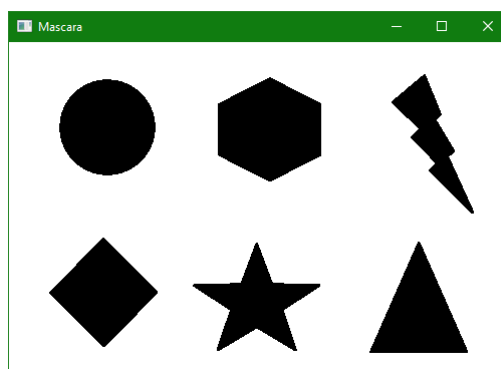


Figura 12: Mascara de imagen del ejercicio 4


```
# Se importa la libreria opencv
import cv2

# Se lee la imagen 'ejercicio5.png'
imagen = cv2.imread("ejercicio5.png")
# Se cambia el tamaño de la imagen para ser vista de mejor manera
imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
# Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva
variable
imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
# Se obtiene la mascara de la imagen
mask = cv2.inRange(imagen_hsv, (58, 255, 238), (58, 255, 238))

# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara de la imagen
cv2.imshow("Mascara", mask)
# OpenCV espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 5 primero se importa la librería **opencv**. Luego se lee la imagen "ejercicio1.png" que se encuentra en el mismo directorio del algoritmo, después se modifica su tamaño para poder ver la imagen de mejor manera y se cambia su formato de BGR a HSV. Por último se calcula la máscara de la imagen y se muestra la imagen original y la máscara de esta. Al final **opencv** espera a que el usuario presione una tecla para terminar el programa.

```
1 # Se importa la libreria opencv
2 import cv2
3
4 # Se lee la imagen 'ejercicio5.png'
5 imagen = cv2.imread("ejercicio5.png")
6 # Se cambia el tamaño de la imagen para ser vista de mejor manera
7 imagen = cv2.resize(imagen, None, fx=0.5, fy=0.5)
8 # Se cambia el formato de la imagen original de BGR a HSV y se asigna a una nueva variable
9 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
10 # Se obtiene la mascara de la imagen
11 mask = cv2.inRange(imagen_hsv, (58, 255, 238), (58, 255, 238))
12
13 # Se muestra la imagen original
14 cv2.imshow("Imagen", imagen)
15 # Se muestra la mascara de la imagen
16 cv2.imshow("Mascara", mask)
17 # Opencv espera a que el usuario presione una tecla
18 cv2.waitKey(0)
19
```

Figura 13: Imagen del código del ejercicio 5



Figura 14: Imagen original del ejercicio 5

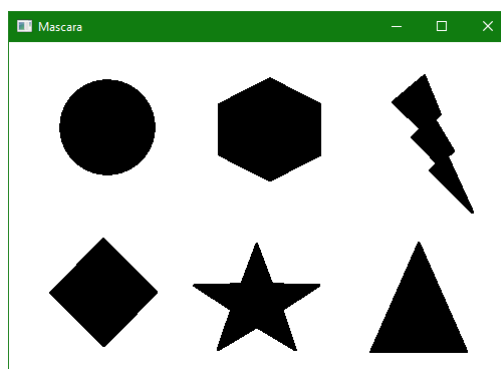


Figura 15: Mascara de imagen del ejercicio 5

```
# Se importa la libreria opencv
import cv2

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

# Se cambia el tamaño de la imagen 'sonic' para ser vista de mejor manera
sonic = cv2.resize(sonic, None, fx=0.3, fy=0.3)
# Se cambia el tamaño de la imagen 'gato' para ser vista de mejor manera
gato = cv2.resize(gato, None, fx=0.45, fy=0.45)
# Se cambia el tamaño de la imagen 'explo' para ser vista de mejor manera
explo = cv2.resize(explo, None, fx=0.2, fy=0.2)

# Se cambia el formato de la imagen 'sonic' de BGR a HSV y se asigna a una nueva
variable
sonic_hsv = cv2.cvtColor(sonic, cv2.COLOR_BGR2HSV)
# Se cambia el formato de la imagen 'gato' de BGR a HSV y se asigna a una nueva
variable
gato_hsv = cv2.cvtColor(gato, cv2.COLOR_BGR2HSV)
# Se cambia el formato de la imagen 'explo' de BGR a HSV y se asigna a una nueva
variable
explo_hsv = cv2.cvtColor(explo, cv2.COLOR_BGR2HSV)

# Se obtiene la mascara de la imagen 'sonic'
sonic_mask = cv2.inRange(sonic_hsv, (56, 240, 245), (65, 255, 255))
# Se obtiene la mascara de la imagen 'gato'
gato_mask = cv2.inRange(gato_hsv, (59, 200, 200), (61, 255, 255))
# Se obtiene la mascara de la imagen 'explo'
explo_mask = cv2.inRange(explo_hsv, (56, 160, 160), (64, 255, 255))

# Se muestra la imagen 'sonic'
cv2.imshow("Sonic", sonic)
# Se muestra la mascara de la imagen 'sonic'
cv2.imshow("Sonic (Mascara)", sonic_mask)

# Se muestra la imagen 'gato'
cv2.imshow("Gato", gato)
# Se muestra la mascara de la imagen 'gato'
cv2.imshow("Gato (Mascara)", gato_mask)

# Se muestra la imagen 'explo'
cv2.imshow("Explosion", explo)
# Se muestra la mascara de la imagen 'explo'
```

```
cv2.imshow("Explosion (Mascara)", explo_mask)
# Opencv espera a que el usuario presione una tecla
cv2.waitKey(0)
```

Para el código del ejercicio 2 con pantalla verde, primero se importa la librería *opencv* y se leen las imágenes "sonic.jpg", "gato.jpg" y "explo.jpg". Luego se cambia el tamaño de cada imagen para poder verlas bien y se cambia su formato de BGR a HSV. Después se calcula la máscara de cada una de las imágenes y se muestra cada imagen con su máscara. Al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

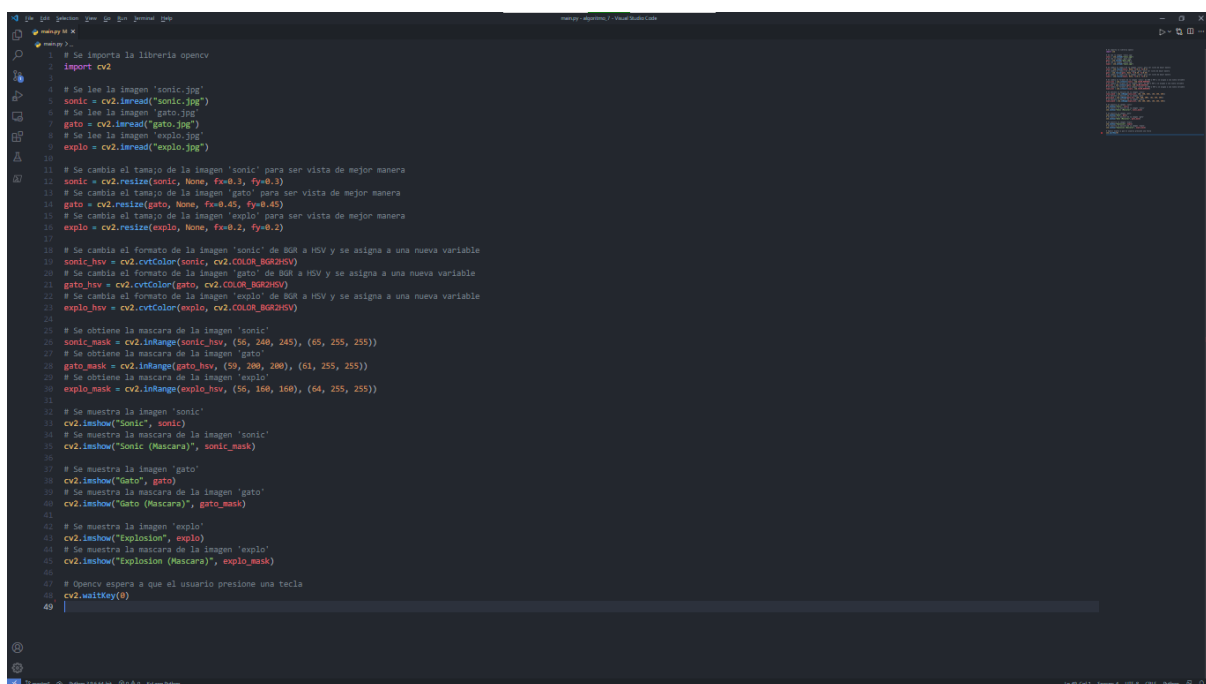


Figura 16: Imagen del código del ejercicio 2 (Greenscreen)

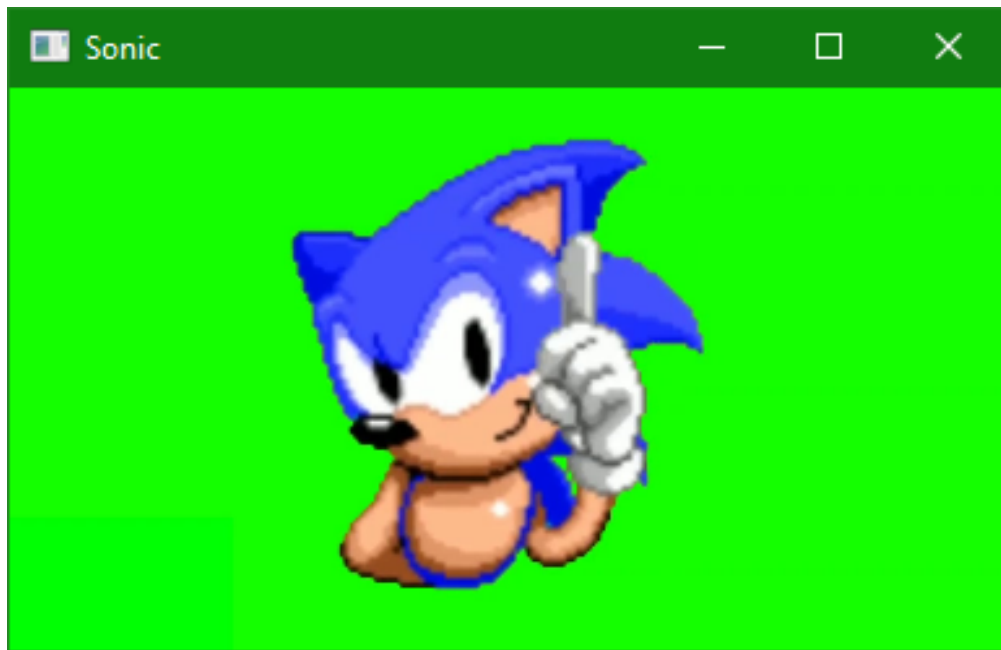


Figura 17: Imagen 'Sonic' original



Figura 18: Mascara de la imagen 'Sonic'

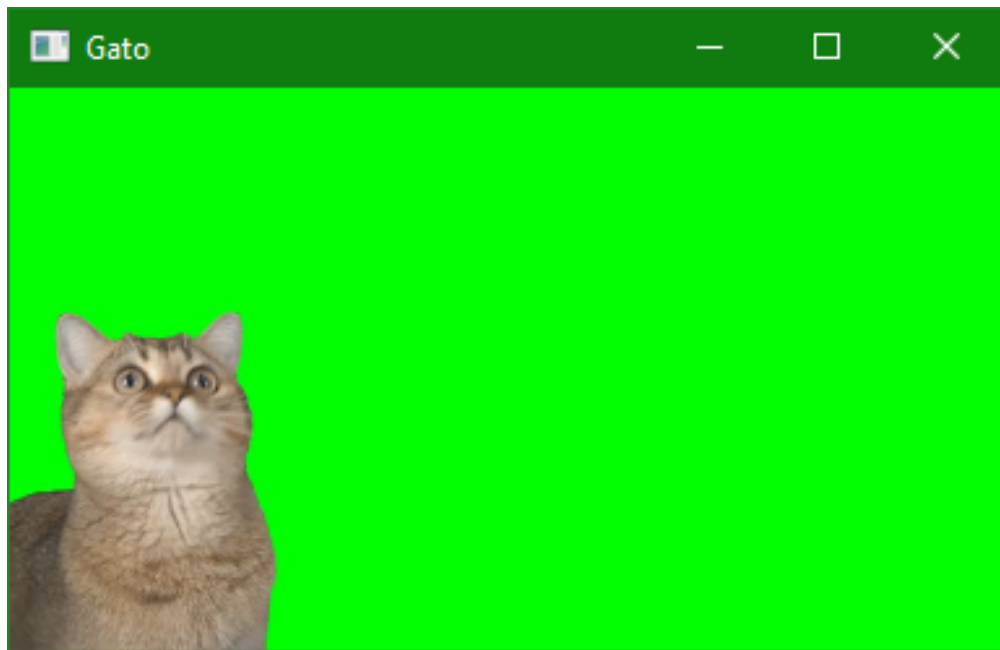


Figura 19: Imagen 'Gato' original

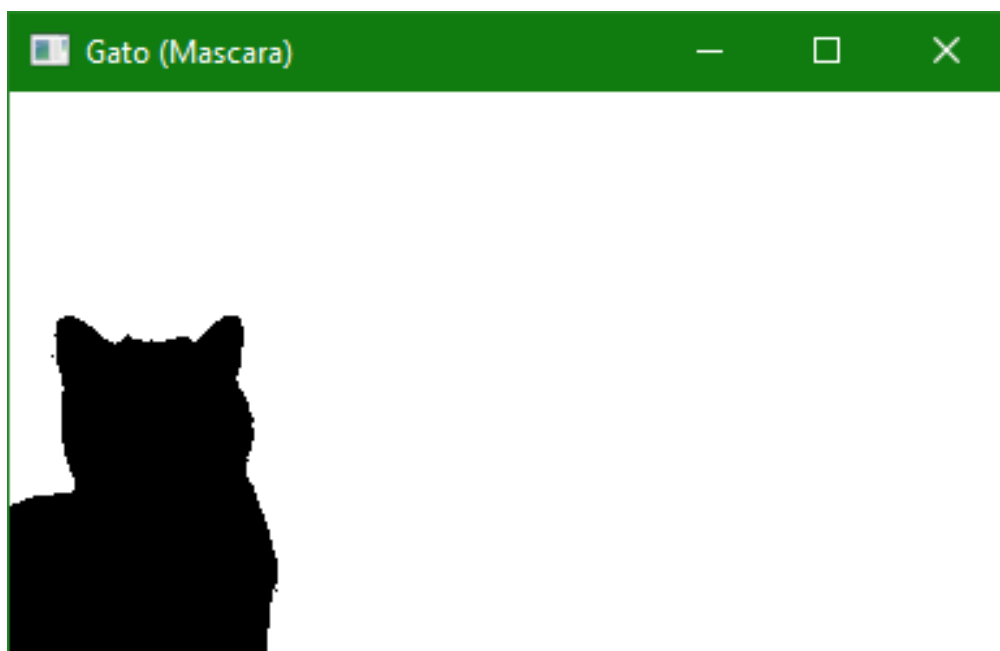


Figura 20: Mascara de la imagen 'Gato'

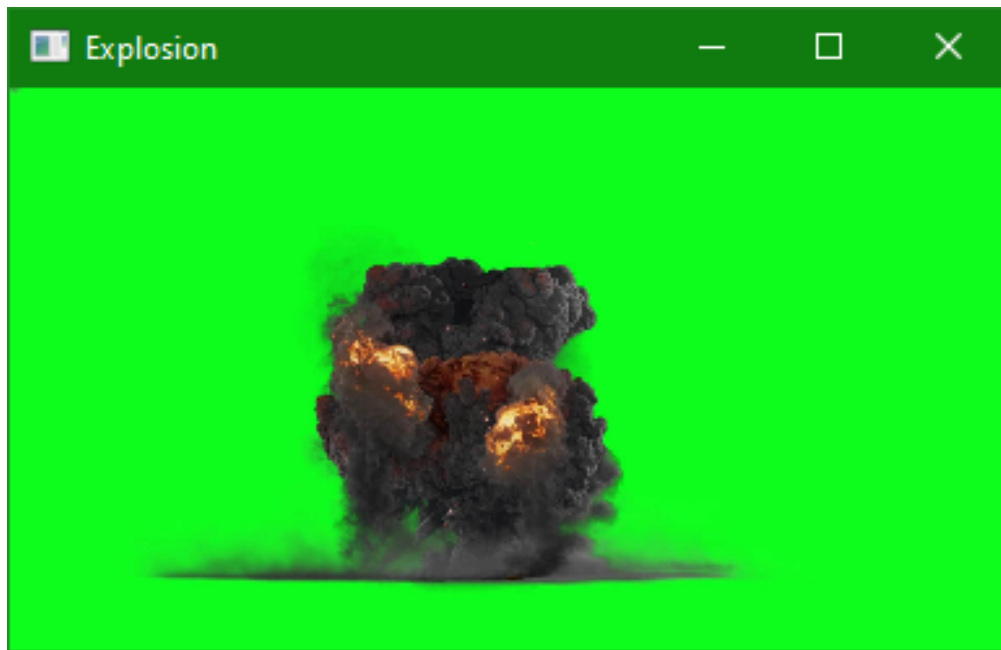


Figura 21: Imagen 'Explosión' original



Figura 22: Mascara de la imagen 'Explosión'