

Universidad de Guadalajara



Algoritmo 13

Muñoz Nuñez Ian Emmanuel

Visión Robótica

"Etiquetado de imágenes"

Índice general

1. Recursos de imagen	2
2. Archivo recurso "etiquetado.py"	3
2.1. Código	3
2.2. Capturas del código del archivo " <i>etiquetado.py</i> "	5
3. Ejercicio 1	6
3.1. Código	6
3.2. Capturas del código del ejercicio 1	8
3.3. Imágenes obtenidas	9
4. Ejercicio 2	11
4.1. Código	11
4.2. Capturas del código del ejercicio 2	13
4.3. Imágenes obtenidas	14

Capítulo 1

Recursos de imagen



Figura 1.1: Recurso imagen 1

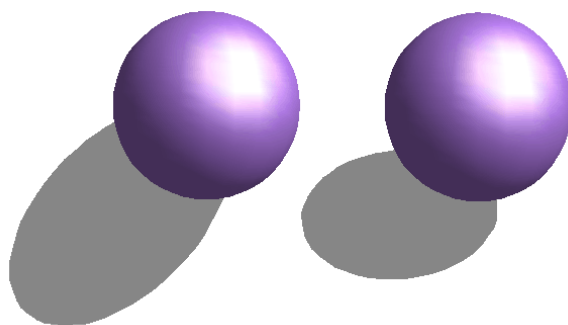


Figura 1.2: Recurso de imagen 2

Capítulo 2

Archivo recurso "etiquetado.py"

2.1. Código

```
# Se importa la libreria opencv
import cv2
# Se importa la libreria numpy como np
import numpy as np

# Se declara la funcion 'etiquetas' para etiquetar la imagen que se recibe
def etiquetas(imagen, numero_objeto):
    # Se cambia el tipo de dato de la imagen de 'uint8' a 'int32'
    imagen = imagen.astype(np.int32)

    # Se obtiene la altura de la imagen
    height = imagen.shape[0]
    # Se obtiene el ancho de la imagen
    width = imagen.shape[1]

    # Se convierten todos los pixels con valor de 255 a -1
    for i in range(height):
        for j in range(width):
            if imagen[i, j] == 255:
                imagen[i, j] = -1

    # Se etiquetan los objetos de la imagen
    cont = 0
    for i in range(height):
        for j in range(width):
            if imagen[i, j] == -1:
                cont += 1
                cv2.floodFill(imagen, None, (j, i), cont)

    # Se obtiene la mascara de la etiqueta que tenga el mismo numero
    # que se recibe en la funcion
    mask = imagen == numero_objeto
```

```
# Se copia la imagen a una nueva variable llamada 'etiqueta'
etiqueta = imagen.copy()
# Se aplica la mascara
etiqueta[mask] = 255
etiqueta[ mask] = 0

# Se cambian los tipos de dato de 'etiqueta' e 'imagen' de 'int32' a 'uint8'
imagen = imagen.astype(np.uint8)
etiqueta = etiqueta.astype(np.uint8)

# Se regresa la imagen 'etiqueta' y la variable 'cont'
return etiqueta, cont
```

El código anterior se encuentra en el archivo ***"etiquetado.py"***, en el archivo se creo una función para ser importada desde el archivo principal. En esta función se reciben dos parametros, la imagen que se quiere etiquetar, y el número de etiqueta del objeto que se quiere ver. En la función primero se convierte el tipo de dato de la imagen de *"uint8"* a ***"int32"***, luego se obtiene la altura y el ancho de la imagen, se convierten todos los pixels con valor de 255 a -1 y se comienza el etiquetado de los objetos, después se calcula la máscara de la imagen y se aplica a una nueva variable, al final se convierten los tipos de dato de la imagen de nuevo a ***"uint8"*** y se regresa la imagen etiquetada junto con el numero de etiquetas que se hicieron.

2.2. Capturas del código del archivo *"etiquetado.py"*

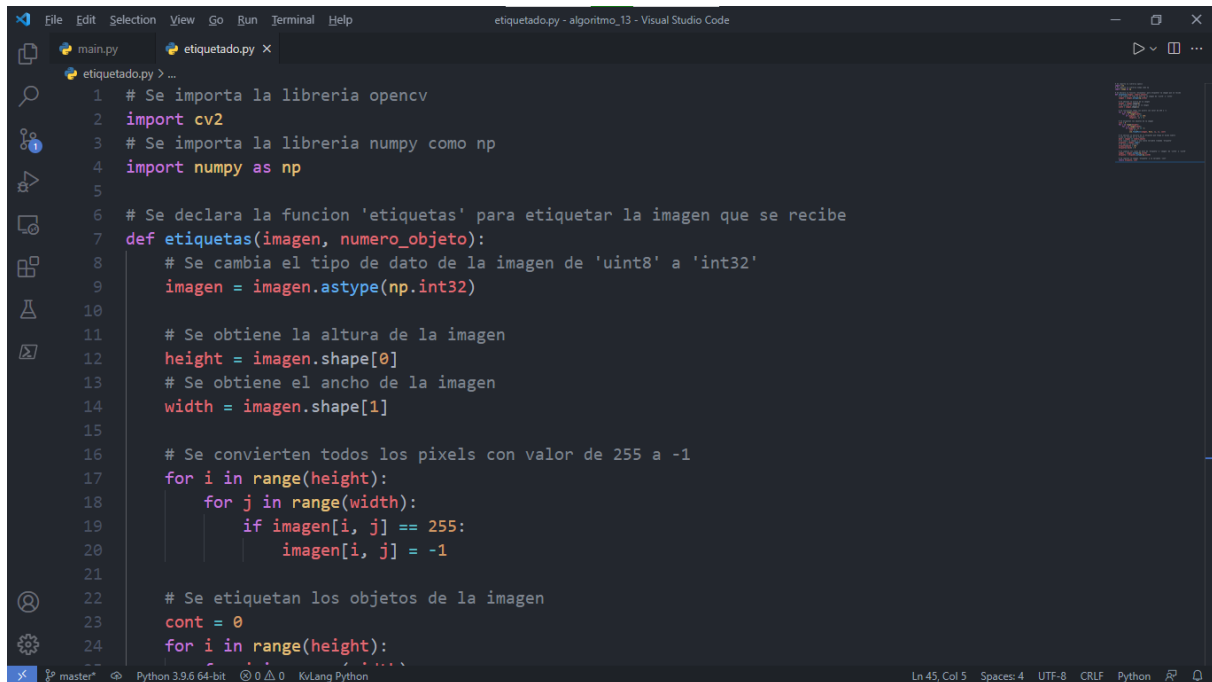


Figura 2.1: Captura 1 del código del archivo *"etiquetado.py"*

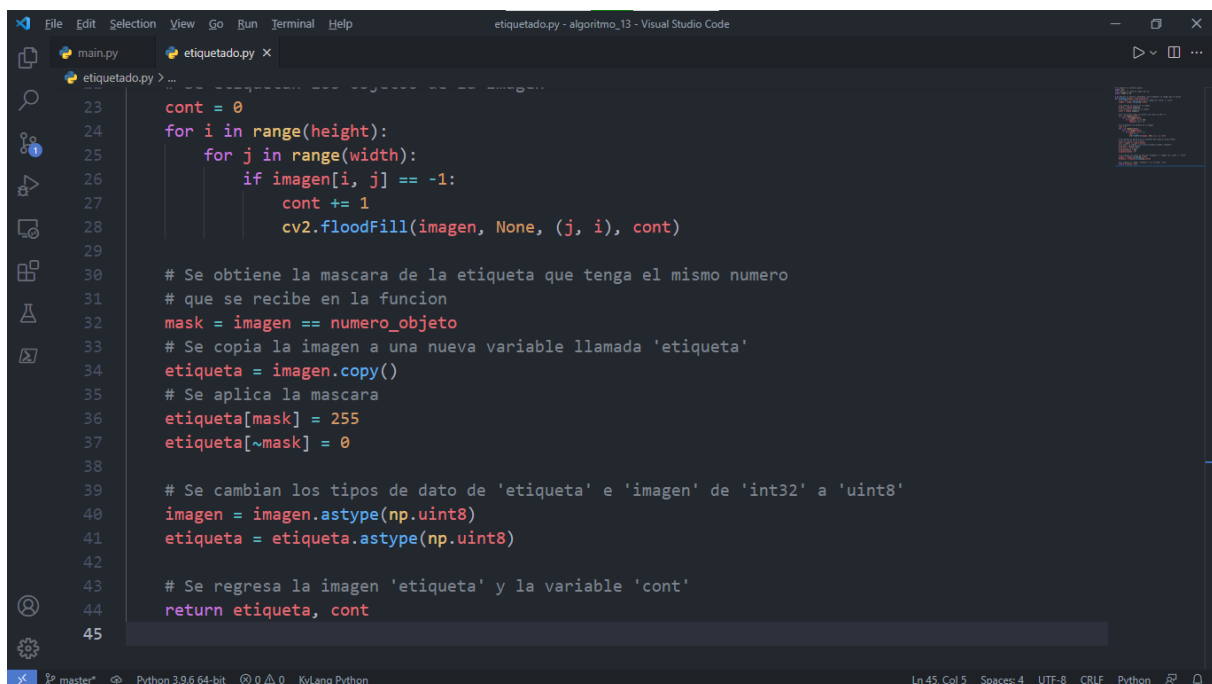


Figura 2.2: Captura 2 del código del archivo *"etiquetado.py"*

Capítulo 3

Ejercicio 1

3.1. Código

```
# Se importa la libreria opencv
import cv2
# Se importa la funcion 'etiquetas' del archivo 'etiquetas.py'
from etiquetado import etiquetas

# Se carga la imagen 'imagen.png'
imagen = cv2.imread("imagen.png")
# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.6, fy=0.6)

# 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
mask = cv2.inRange(imagen_hsv, (2, 150, 180), (15, 255, 255))
# Se calcula la mascara de la imagen para usarla con la funcion floodfill
floodfill = cv2.inRange(imagen_hsv, (2, 150, 180), (15, 255, 255))

# Se rellena la mascara
cv2.floodFill(floodfill, None, (0, 0), 255)
# Se calcula el negativo del floodfill
floodfill_negativo = cv2.bitwise_not(floodfill)

# Se combinan la mascara el negativo del floodfill fill = cv2.bitwise_or(mask,
floodfill_negativo)

# Se obtienen las etiquetas y la imagen con estas
etiqueta, _ = etiquetas(fill, 0)

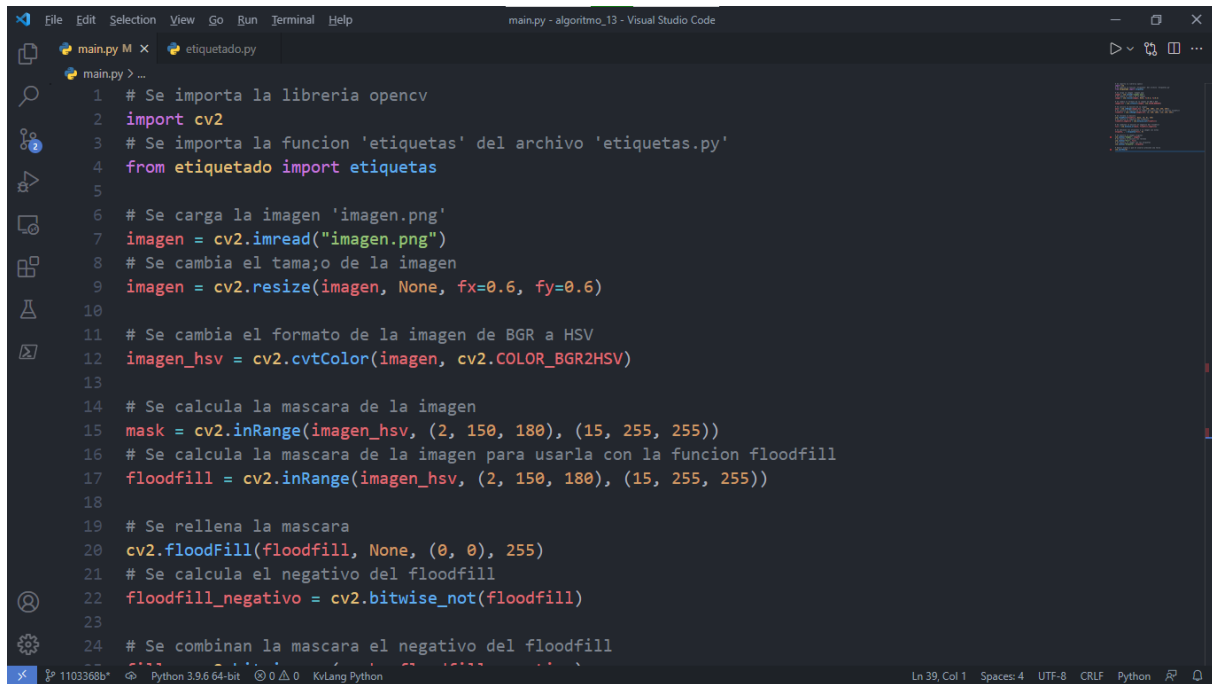
# Se muestra la imagen original
cv2.imshow("Imagen", imagen)
# Se muestra la mascara con relleno
```

```
cv2.imshow("Fill", fill)
# Se muestra la imagen con las etiquetas
cv2.imshow("Etiqueta", etiqueta)

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

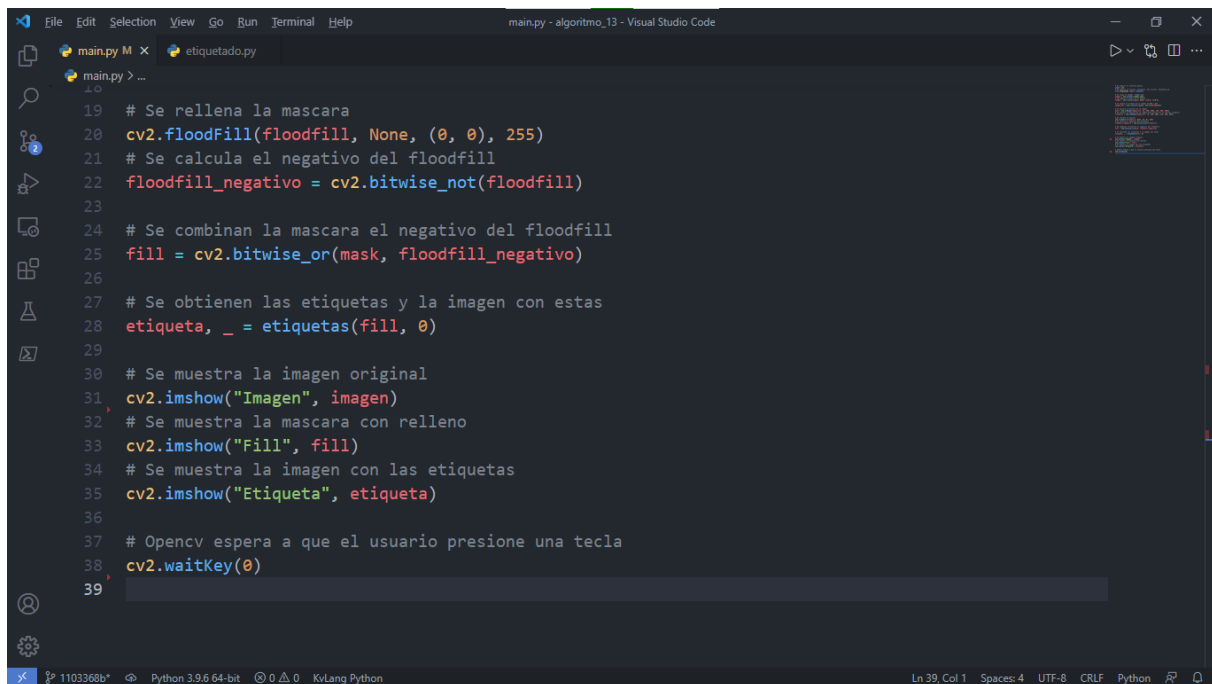
Para el código del ejercicio 1 del algoritmo 13, primero se importa la librería **"opencv"** y la función **"etiquetas"** del archivo **"etiquetado.py"**. Luego se carga la imagen 'imagen.png', se modifica su tamaño y también su formato de BGR a HSV. Después se calcula una máscara para ser usada en el relleno final de la imagen y otra para ser rellenada, se rellena la máscara de esta última y se calcula su negativo. Se combinan la máscara y el negativo del floodfill para obtener una máscara con relleno. Por último se obtiene la imagen con etiquetas y las etiquetas que se contaron con la función **"etiquetas"** y se muestran las imágenes que se obtuvieron de resultado. Al final **"opencv"** espera a que el usuario presione una tecla para terminar el programa.

3.2. Capturas del código del ejercicio 1



```
1 # Se importa la libreria opencv
2 import cv2
3 # Se importa la funcion 'etiquetas' del archivo 'etiquetas.py'
4 from etiquetado import etiquetas
5
6 # Se carga la imagen 'imagen.png'
7 imagen = cv2.imread("imagen.png")
8 # Se cambia el tamaño de la imagen
9 imagen = cv2.resize(imagen, None, fx=0.6, fy=0.6)
10
11 # Se cambia el formato de la imagen de BGR a HSV
12 imagen_hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
13
14 # Se calcula la mascara de la imagen
15 mask = cv2.inRange(imagen_hsv, (2, 150, 180), (15, 255, 255))
16 # Se calcula la mascara de la imagen para usarla con la funcion floodfill
17 floodfill = cv2.inRange(imagen_hsv, (2, 150, 180), (15, 255, 255))
18
19 # Se rellena la mascara
20 cv2.floodFill(floodfill, None, (0, 0), 255)
21 # Se calcula el negativo del floodfill
22 floodfill_negativo = cv2.bitwise_not(floodfill)
23
24 # Se combinan la mascara el negativo del floodfill
```

Figura 3.1: Captura 1 del código del ejercicio 1



```
19 # Se rellena la mascara
20 cv2.floodFill(floodfill, None, (0, 0), 255)
21 # Se calcula el negativo del floodfill
22 floodfill_negativo = cv2.bitwise_not(floodfill)
23
24 # Se combinan la mascara el negativo del floodfill
25 fill = cv2.bitwise_or(mask, floodfill_negativo)
26
27 # Se obtienen las etiquetas y la imagen con estas
28 etiqueta, _ = etiquetas(fill, 0)
29
30 # Se muestra la imagen original
31 cv2.imshow("Imagen", imagen)
32 # Se muestra la mascara con relleno
33 cv2.imshow("Fill", fill)
34 # Se muestra la imagen con las etiquetas
35 cv2.imshow("Etiqueta", etiqueta)
36
37 # Opencv espera a que el usuario presione una tecla
38 cv2.waitKey(0)
39
```

Figura 3.2: Captura 2 del código del ejercicio 2

3.3. Imágenes obtenidas

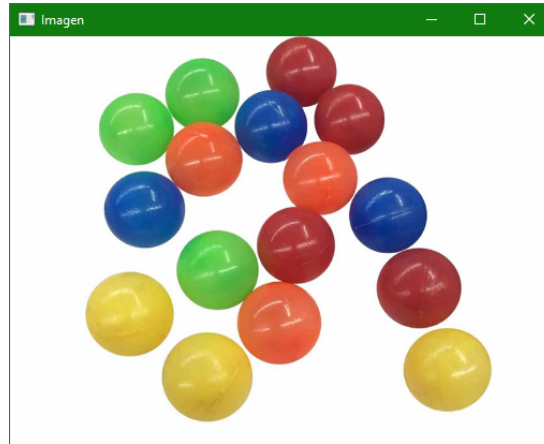


Figura 3.3: Imagen original

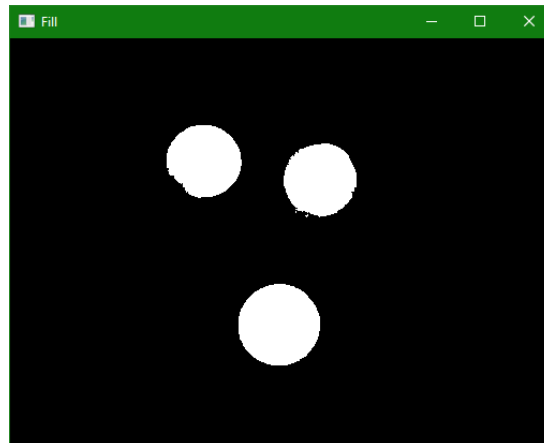


Figura 3.4: Relleno de la imagen

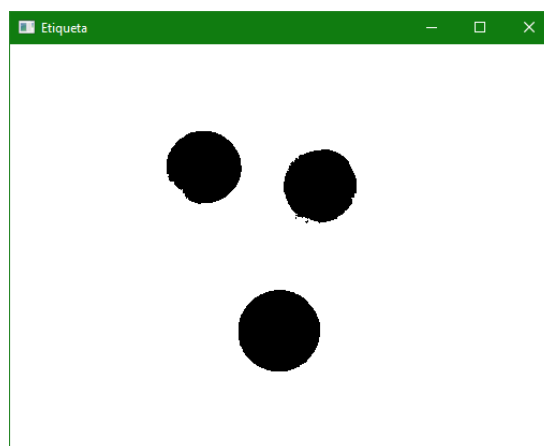


Figura 3.5: Etiqueta 0 de la imagen

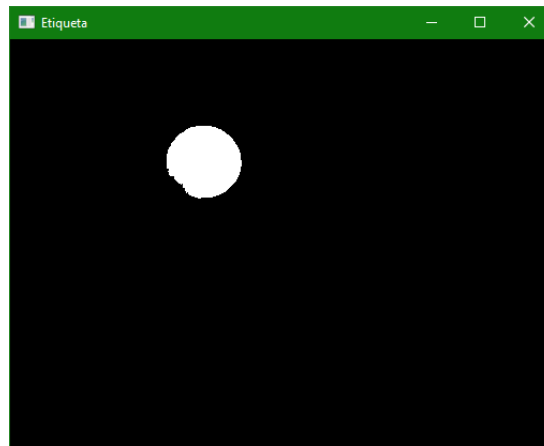


Figura 3.6: Etiqueta 1 de la imagen

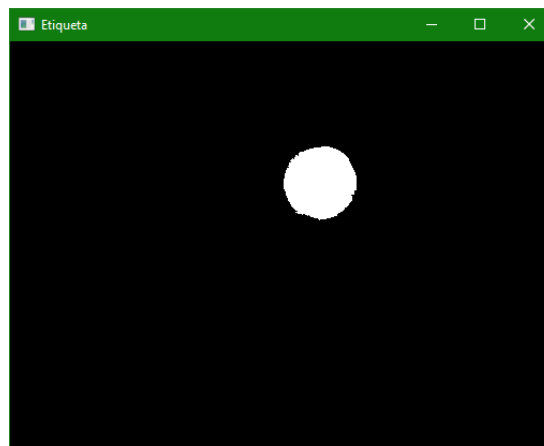


Figura 3.7: Etiqueta 2 de la imagen

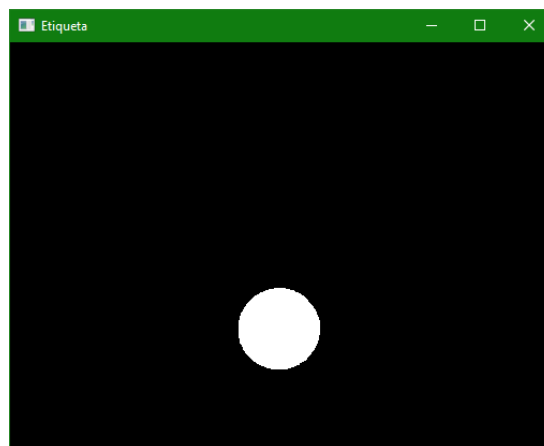


Figura 3.8: Etiqueta 3 de la imagen

Capítulo 4

Ejercicio 2

4.1. Código

```
# Se importa la libreria opencv
import cv2
# Se importa la funcion 'etiquetas' del archivo 'etiquetas.py'
from etiquetado import etiquetas

# Se carga la imagen 'esferas.png'
imagen = cv2.imread("esferas.png")
# Se cambia el tamaño de la imagen
imagen = cv2.resize(imagen, None, fx=0.8, fy=0.8)

# 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
mask = cv2.inRange(imagen_hsv, (130, 100, 80), (140, 255, 255))
# Se calcula la mascara de la imagen para usarla con la funcion floodfill
floodfill = cv2.inRange(imagen_hsv, (130, 100, 80), (140, 255, 255))

# Se rellena la mascara
cv2.floodFill(floodfill, None, (0, 0), 255)
# Se calcula el negativo del floodfill
floodfill_negativo = cv2.bitwise_not(floodfill)

# Se combinan la mascara el negativo del floodfill
fill = cv2.bitwise_or(mask, floodfill_negativo)

# Se obtienen las etiquetas y la imagen con estas
etiqueta, cont = etiquetas(fill, 0)

print(cont)

# Se muestra la imagen original
```

```
cv2.imshow("Imagen", imagen)
# Se muestra la mascara
cv2.imshow("Mascara", mask)
# Se muestra el floodfill
cv2.imshow("Floodfill", floodfill)
# Se muestra el negativo del floodfill
cv2.imshow("Floodfill (Negativo)", floodfill_negativo)
# Se muestra la mascara con relleno
cv2.imshow("Fill", fill)
# Se muestra la imagen con las etiquetas
cv2.imshow("Etiqueta", etiqueta)

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

Para el código del ejercicio 2 del algoritmo 13, primero se importa la librería **"opencv"** y la función **"etiquetas"** del archivo **"etiquetado.py"**. Luego se carga la imagen 'esferas.png', se modifica su tamaño y también su formato de BGR a HSV. Después se calcula una máscara para ser usada en el relleno final de la imagen y otra para ser rellenada, se rellena la máscara de esta última y se calcula su negativo. Se combinan la máscara y el negativo del floodfill para obtener una máscara con relleno. Por último se obtiene la imagen con etiquetas y las etiquetas que se contaron con la función **"etiquetas"** y se muestran las imágenes que se obtuvieron de resultado junto con una impresión en pantalla de las etiquetas que se contaron. Al final **"opencv"** espera a que el usuario presione una tecla para terminar el programa.

4.2. Capturas del código del ejercicio 2

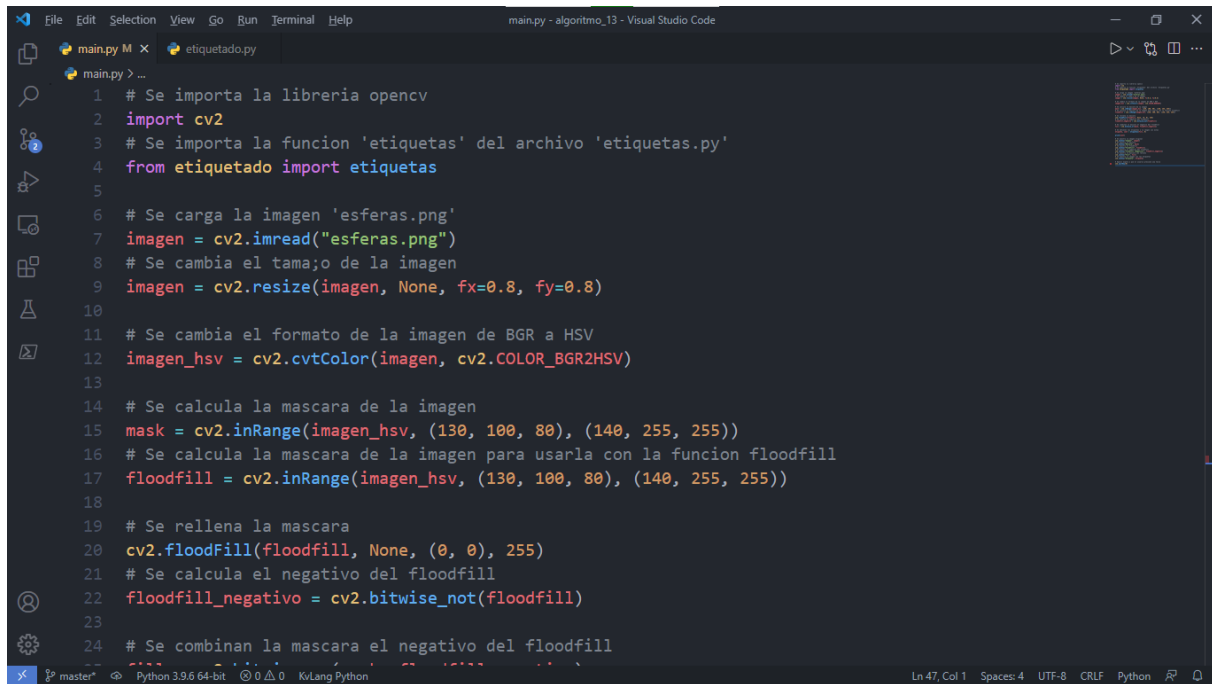


Figura 4.1: Captura 1 del código del ejercicio 1

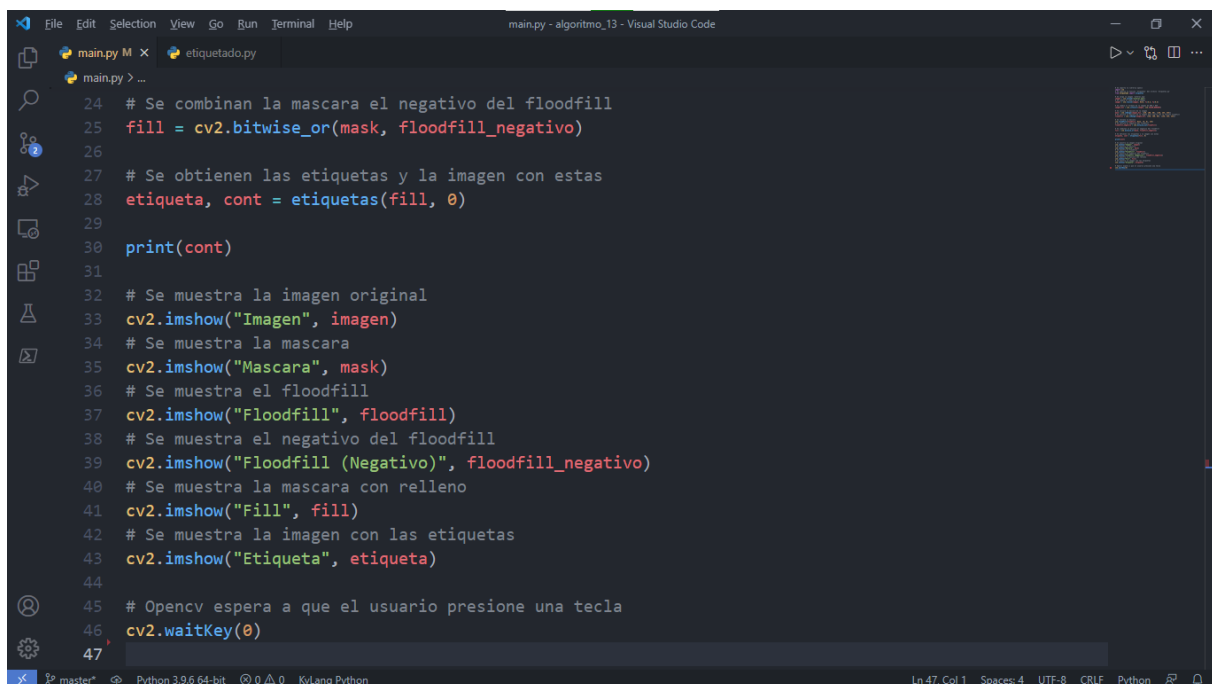


Figura 4.2: Captura 2 del código del ejercicio 2

4.3. Imágenes obtenidas

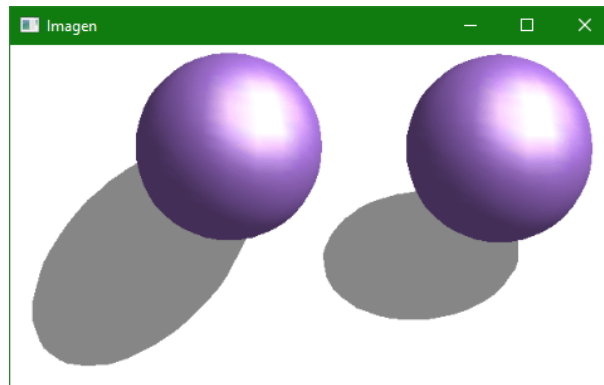


Figura 4.3: Imagen original

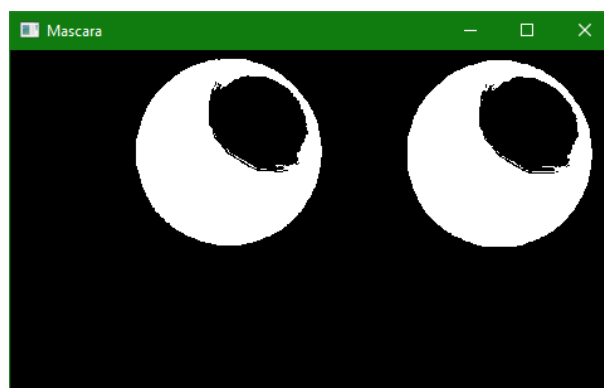


Figura 4.4: Máscara

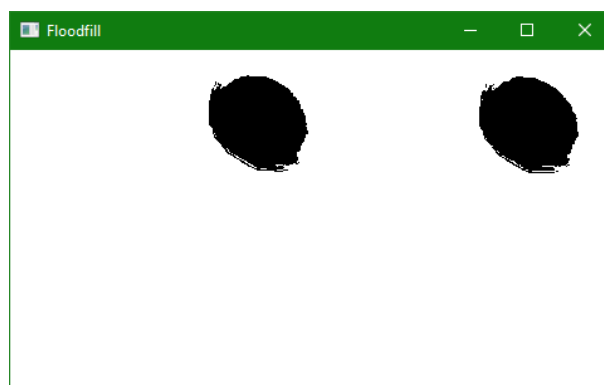


Figura 4.5: Floodfill



Figura 4.6: Negativo del floodfill

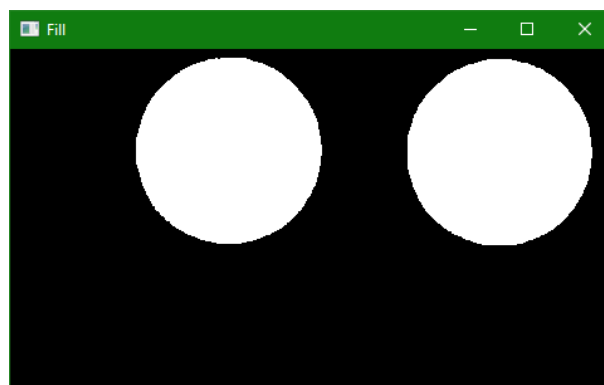


Figura 4.7: Fill

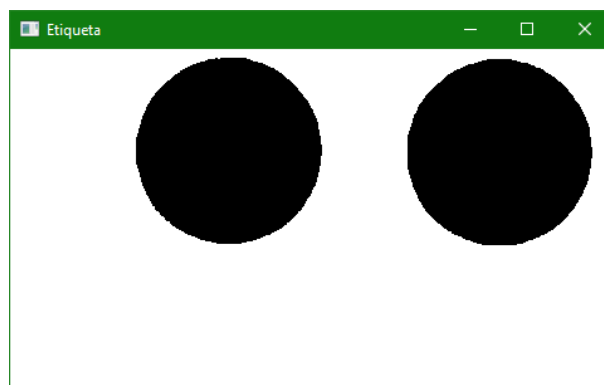


Figura 4.8: Etiqueta 0 de la imagen

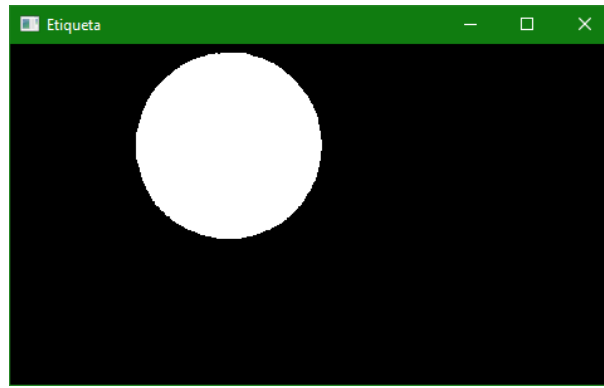


Figura 4.9: Etiqueta 1 de la imagen

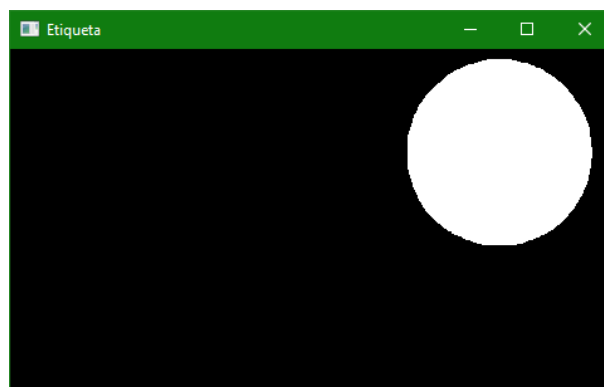


Figura 4.10: Etiqueta 2 de la imagen

Si se busca una etiqueta nula en ambos ejercicios se muestra la siguiente imagen.

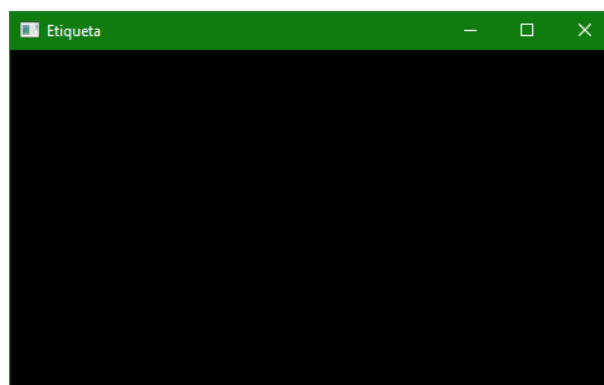


Figura 4.11: Resultado de buscar una etiqueta nula