

Algoritmo 6

Muñoz Nuñez Ian Emmanuel

Modelo HSV

Visión Robótica

Código del punto 1 de los ejercicios Hue

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 1 de los ejercicios Hue primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help main.py - algoritmo_6 - Visual Studio Code
main.py M x
main.py > ...
1 # Se importa la libreria opencv
2 import cv2
3 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 30
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 255
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 255
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 1: Código del punto 1

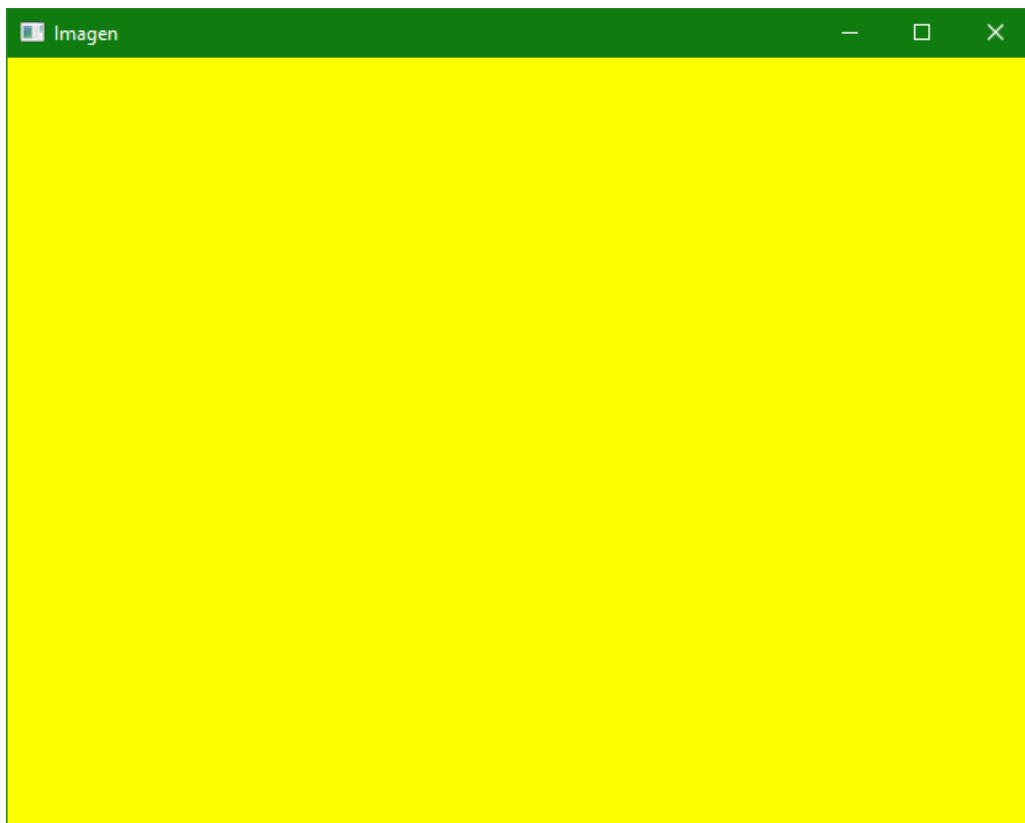


Figura 2: Imagen del punto 1

Código del punto 2 de los ejercicios Hue

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

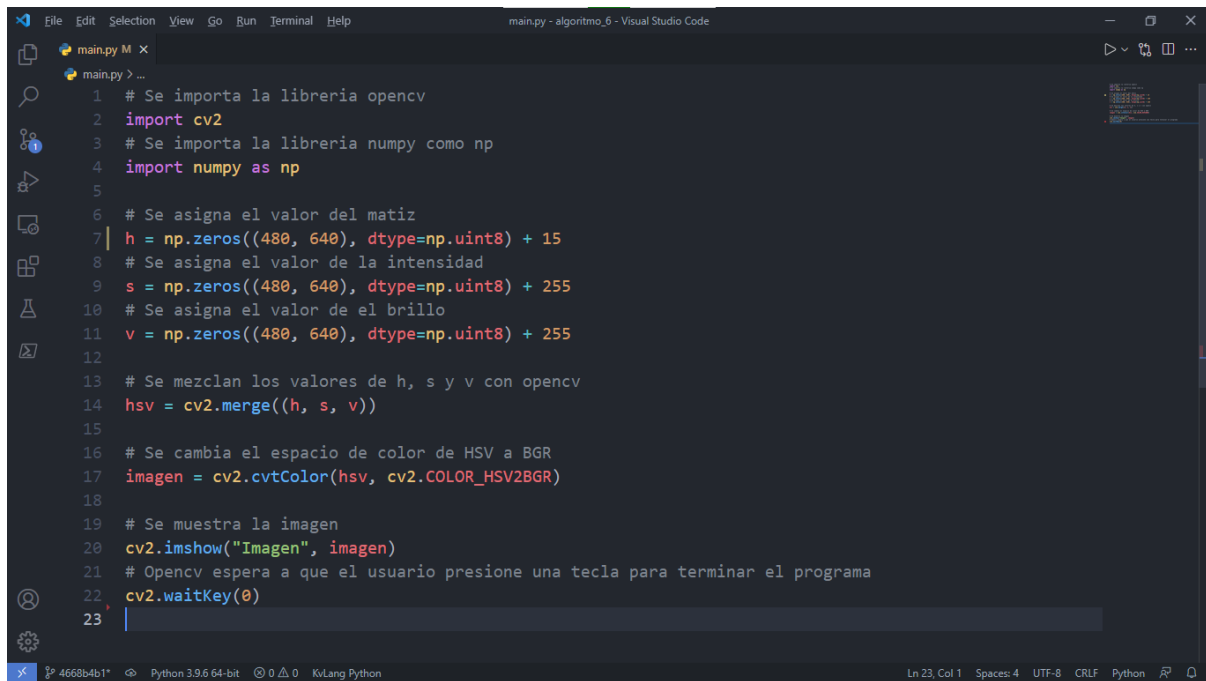
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 15
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 2 de los ejercicios Hue primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, 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 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 15
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 255
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 255
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 3: Código del punto 2

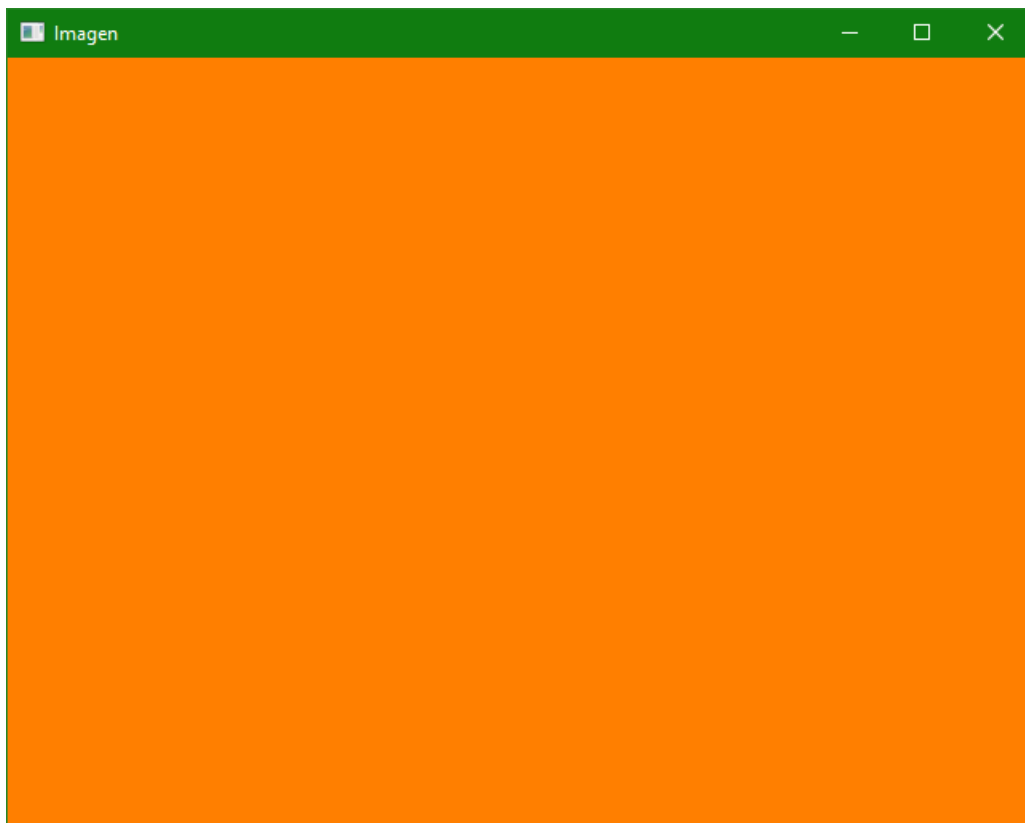


Figura 4: Imagen del punto 2

Código del punto 3 de los ejercicios Hue

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

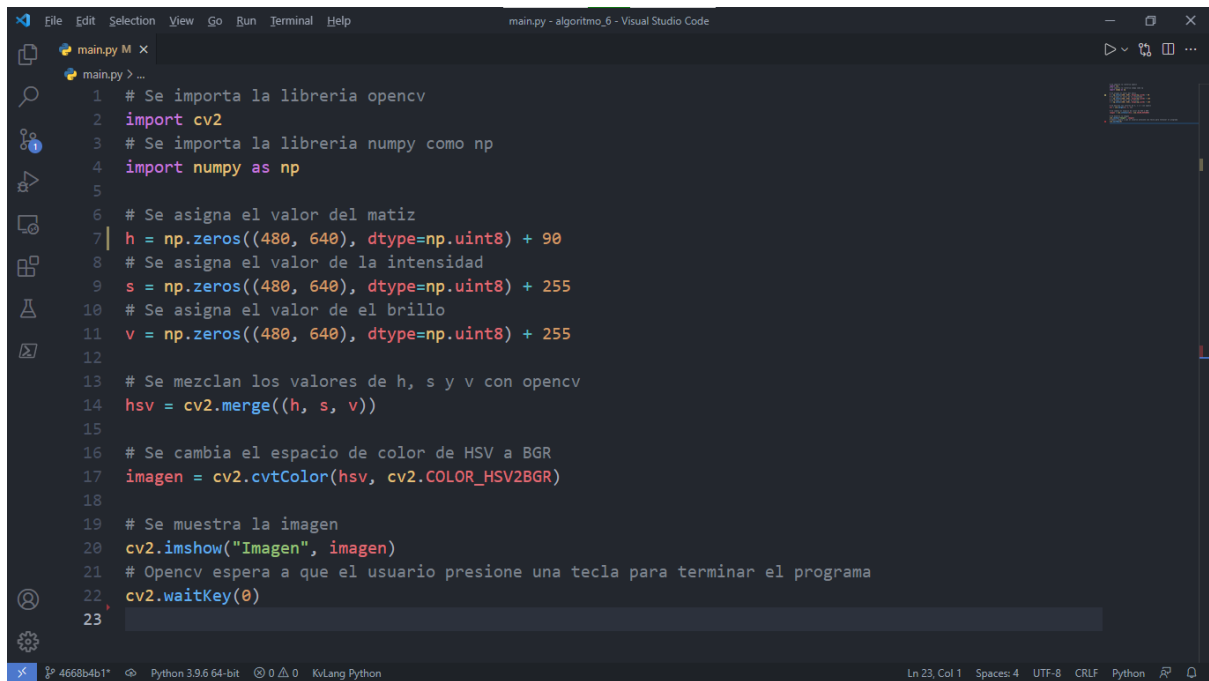
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 90
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 3 de los ejercicios Hue primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, 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 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 90
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 255
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 255
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 5: Código del punto 3

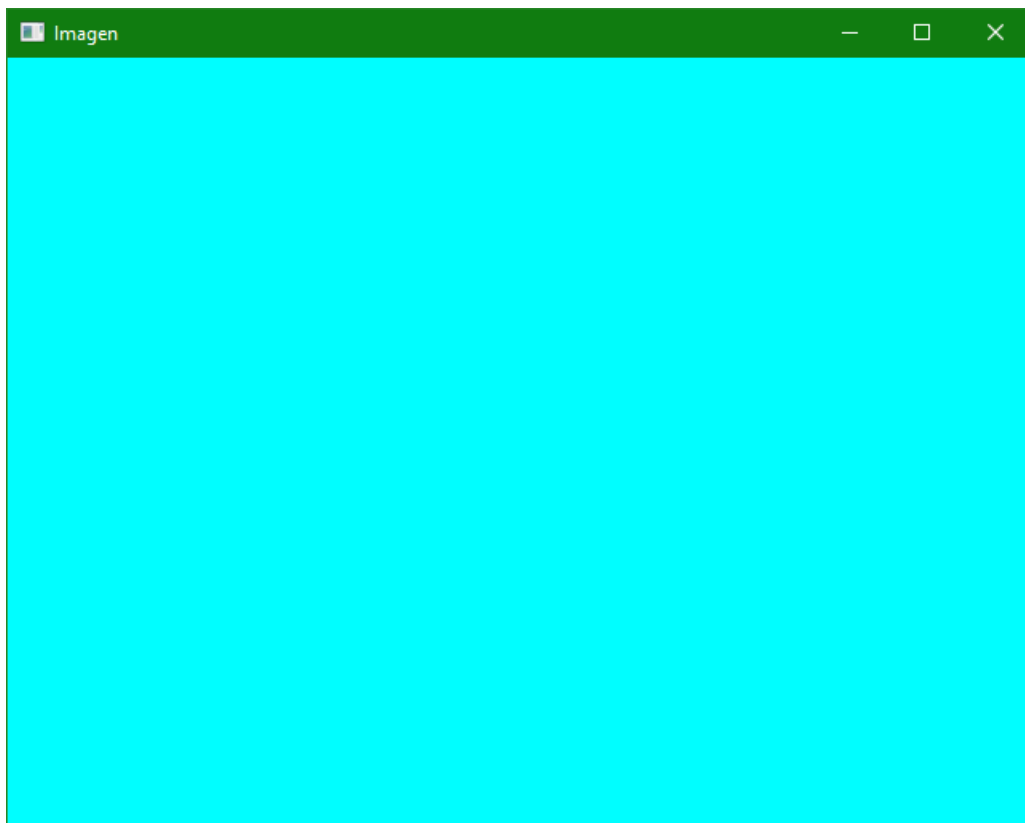


Figura 6: Imagen del punto 3

Código del punto 4 de los ejercicios Hue

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

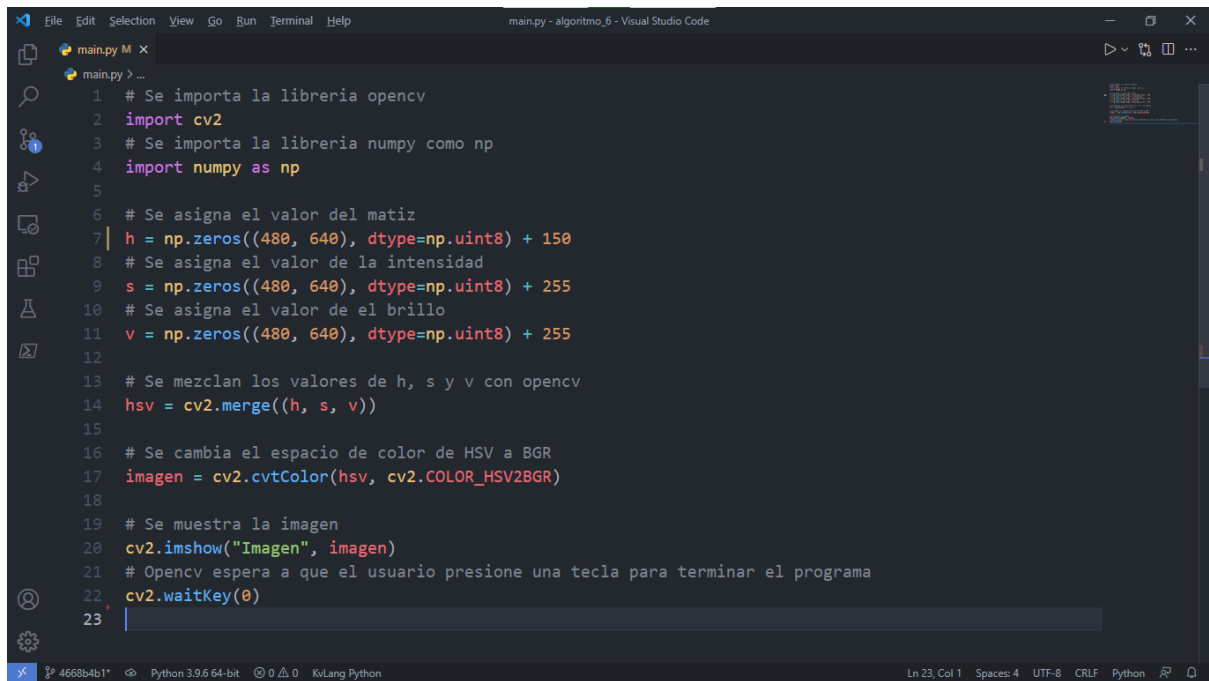
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 150
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 4 de los ejercicios Hue primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, 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 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 150
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 255
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 255
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 7: Código del punto 4

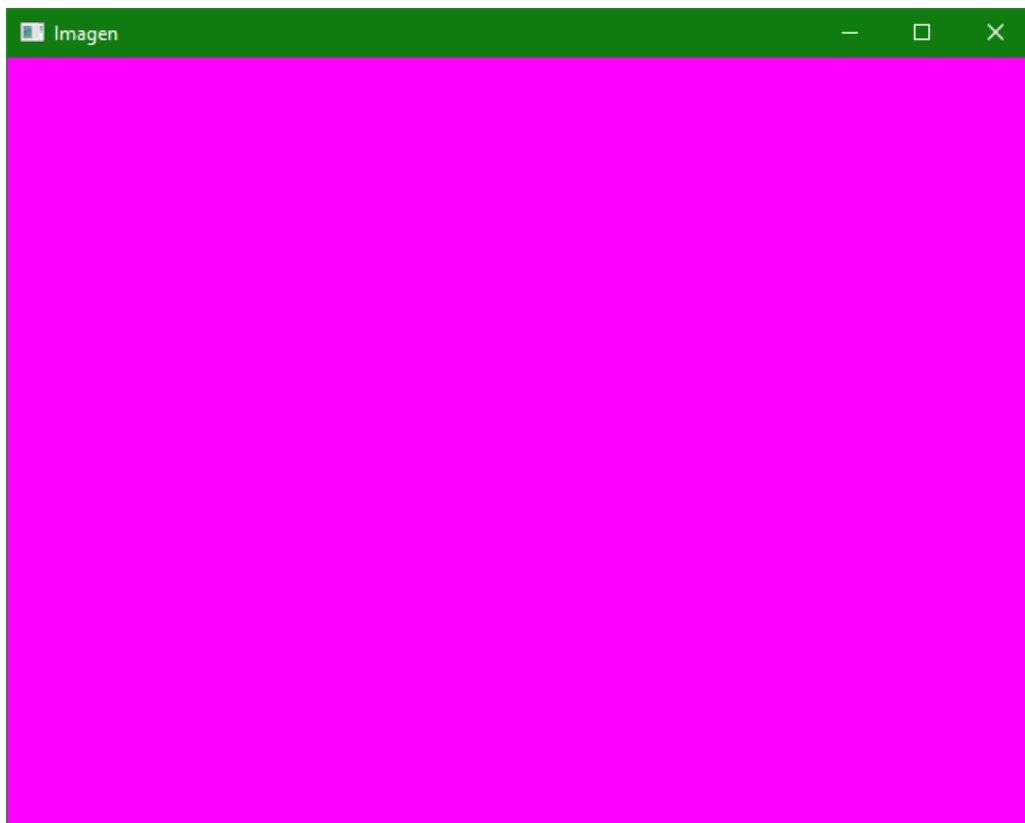


Figura 8: Imagen del punto 4

Código del punto 5 de los ejercicios Hue

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 0
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 5 de los ejercicios Hue primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help
main.py - algoritmo_6 - Visual Studio Code

main.py M x
main.py > ...
1 # Se importa la libreria opencv
2 import cv2
3 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 105
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 255
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 255
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 9: Código del punto 5

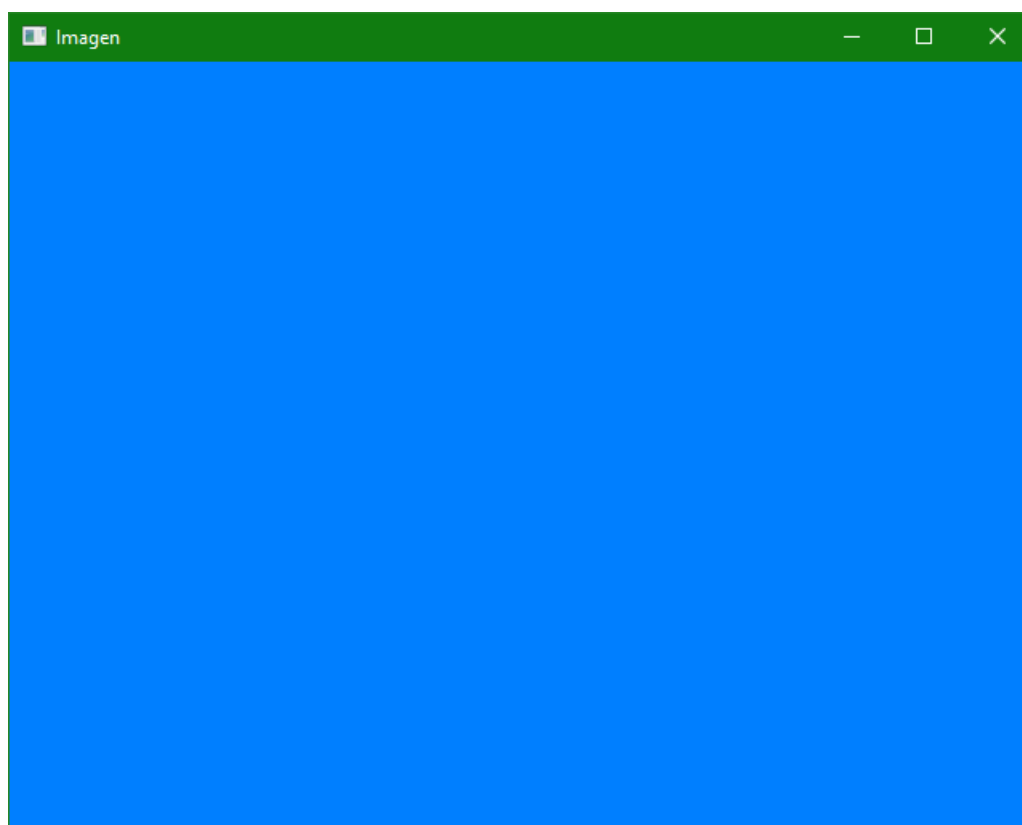


Figura 10: Imagen del punto 5

Código del punto 1 de los ejercicios Saturation

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 1 de los ejercicios Saturation primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help
main.py - algoritmo_6 - Visual Studio Code

main.py M x
main.py > ...
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 255
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 255
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
23
```

Figura 11: Código del punto 1

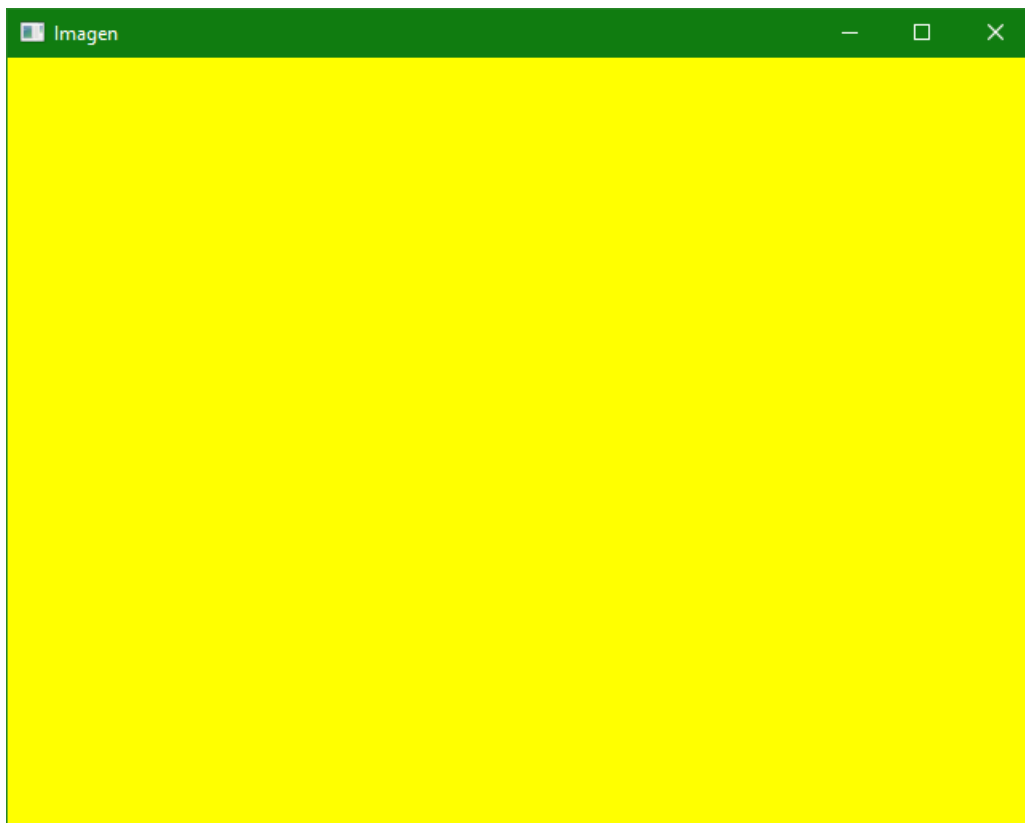


Figura 12: Imagen del punto 1

Código del punto 2 de los ejercicios Saturation

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 204
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 2 de los ejercicios Saturation primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help
main.py - algoritmo_6 - Visual Studio Code

main.py M x
main.py > ...
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 204
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 255
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
23
```

Figura 13: Código del punto 2



Figura 14: Imagen del punto 2

Código del punto 3 de los ejercicios Saturation

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 153
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 3 de los ejercicios Saturation primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help
main.py - algoritmo_6 - Visual Studio Code

main.py M x
main.py > ...
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 153
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 255
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
23
```

Figura 15: Código del punto 3



Figura 16: Imagen del punto 3

Código del punto 4 de los ejercicios Saturation

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 102
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 4 de los ejercicios Saturation primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help
main.py - algoritmo_6 - Visual Studio Code

main.py M x
main.py > ...
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 102
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 255
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
23
```

Figura 17: Código del punto 4

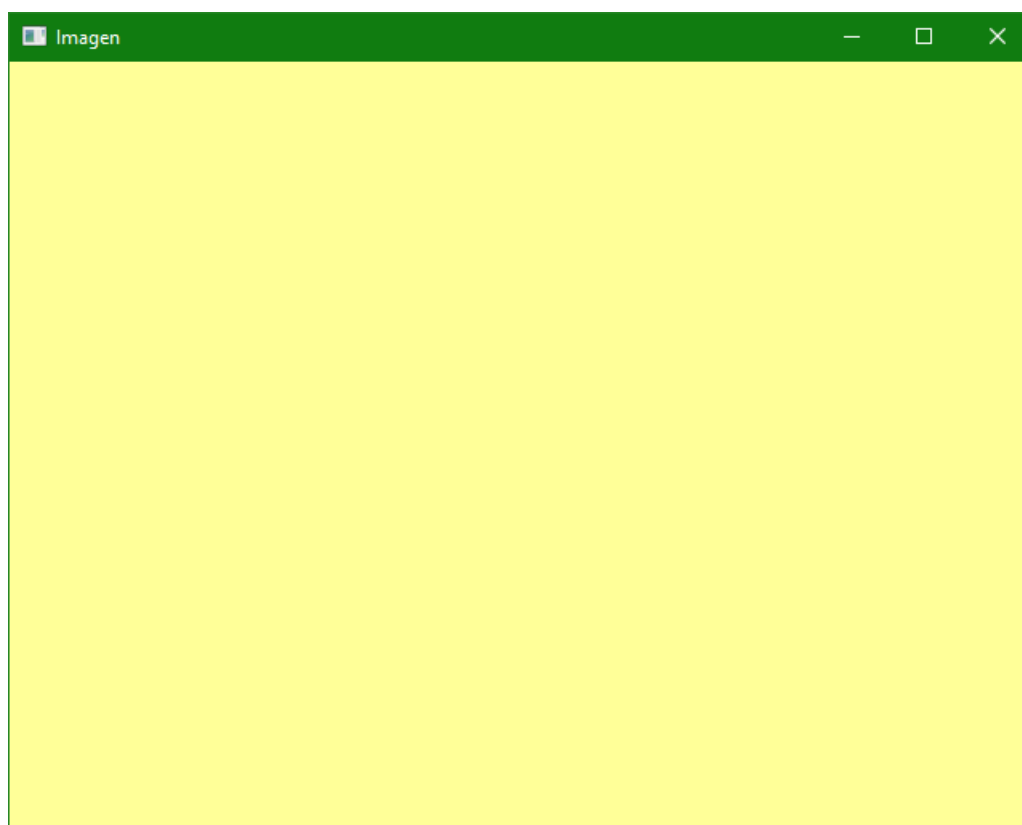


Figura 18: Imagen del punto 4

Código del punto 5 de los ejercicios Saturation

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

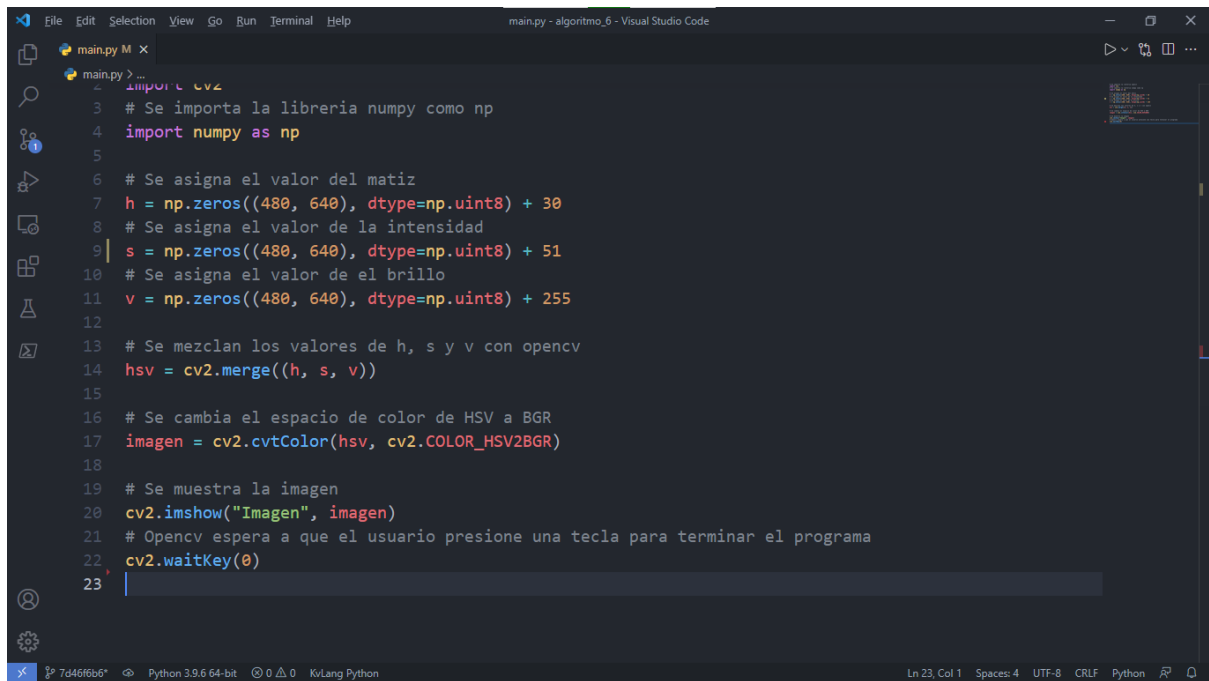
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 51
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 5 de los ejercicios Saturation primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 51
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 255
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Figura 19: Código del punto 5

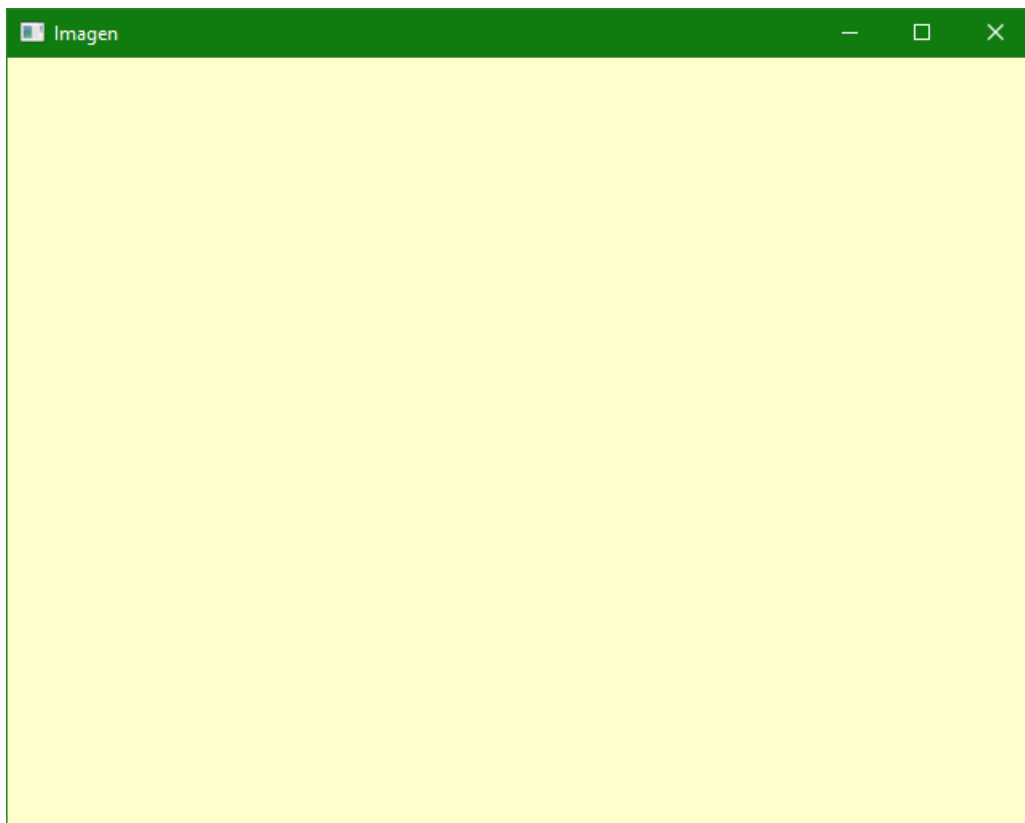


Figura 20: Imagen del punto 5

Código del punto 1 de los ejercicios Value

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

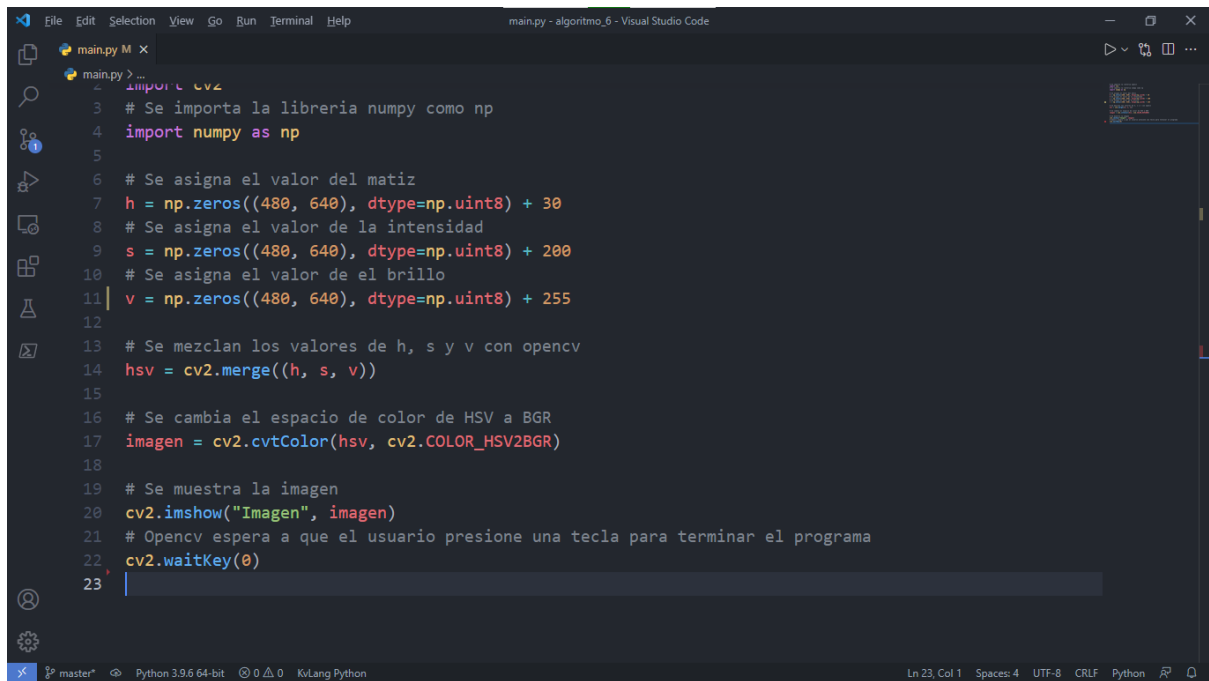
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 200
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 1 de los ejercicios Value primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 200
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 255
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Figura 21: Código del punto 1

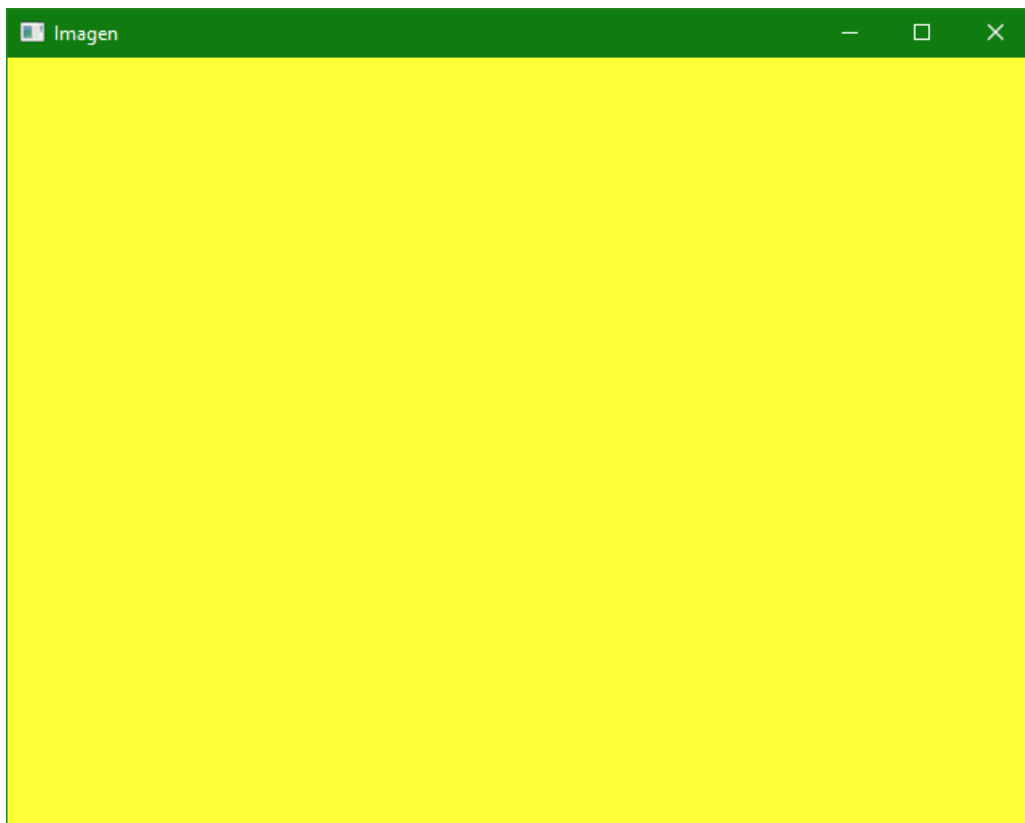


Figura 22: Imagen del punto 1

Código del punto 2 de los ejercicios Value

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

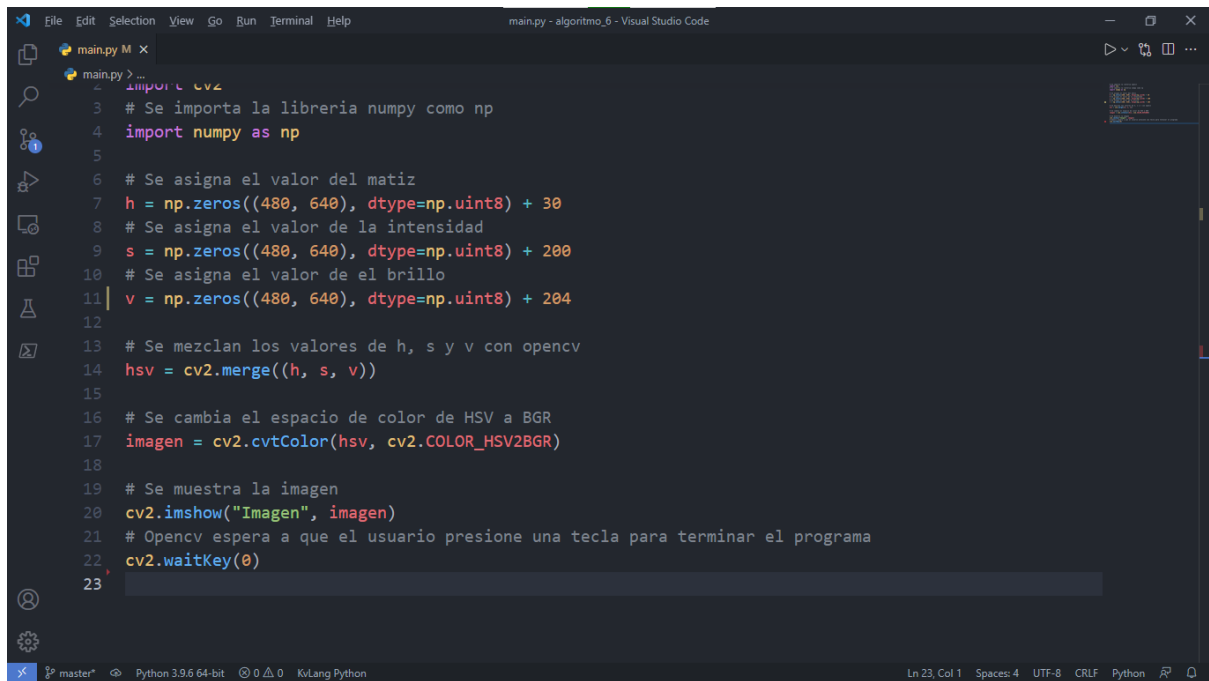
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 200
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 204

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 2 de los ejercicios Value primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 200
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 204
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Figura 23: Código del punto 2

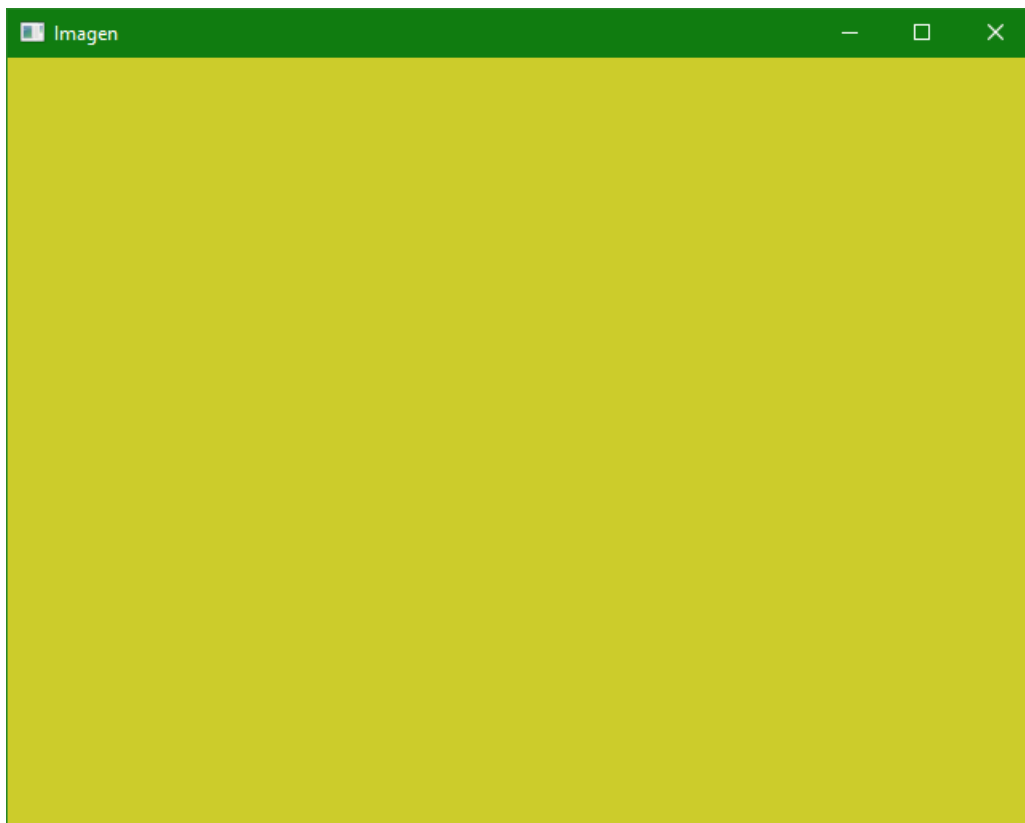


Figura 24: Imagen del punto 2

Código del punto 3 de los ejercicios Value

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

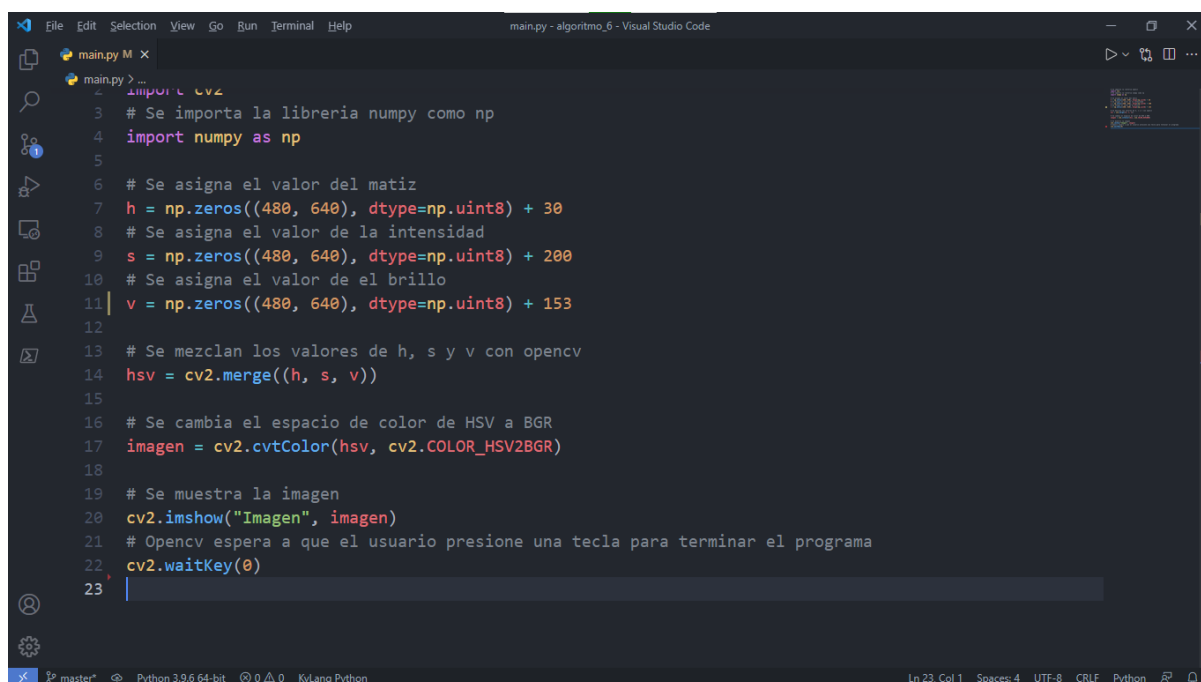
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 200
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 153

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 3 de los ejercicios Value primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 200
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 153
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Figura 25: Código del punto 3

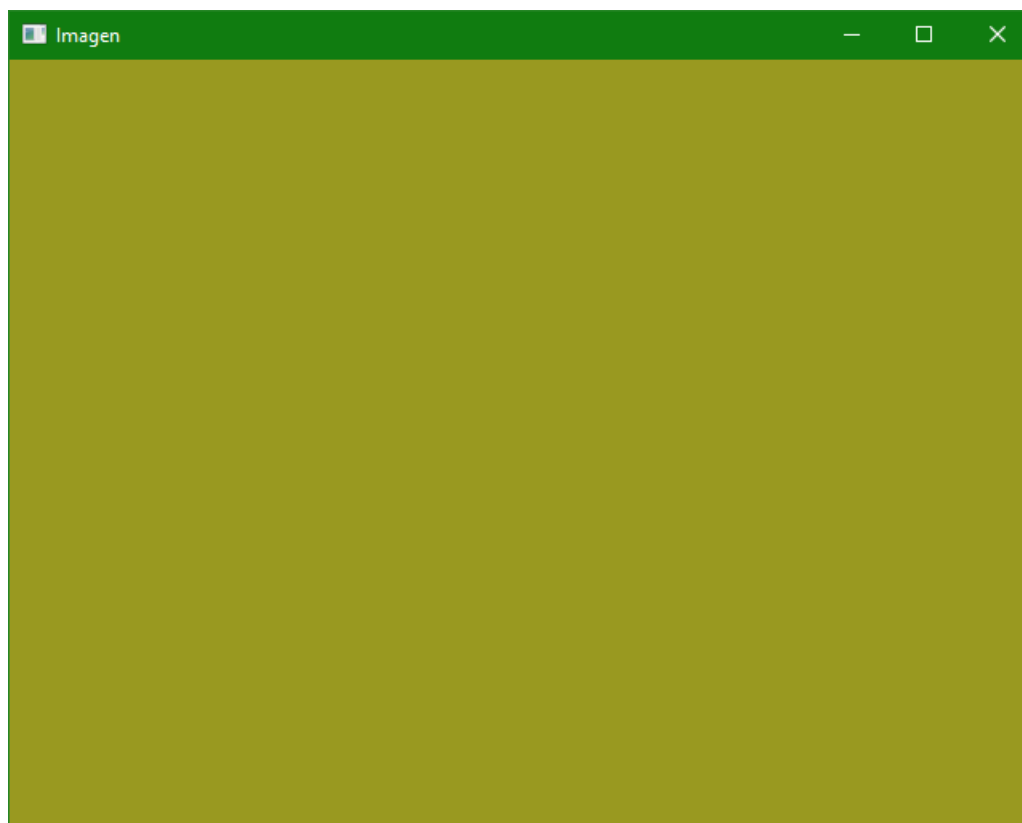


Figura 26: Imagen del punto 3

Código del punto 4 de los ejercicios Value

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

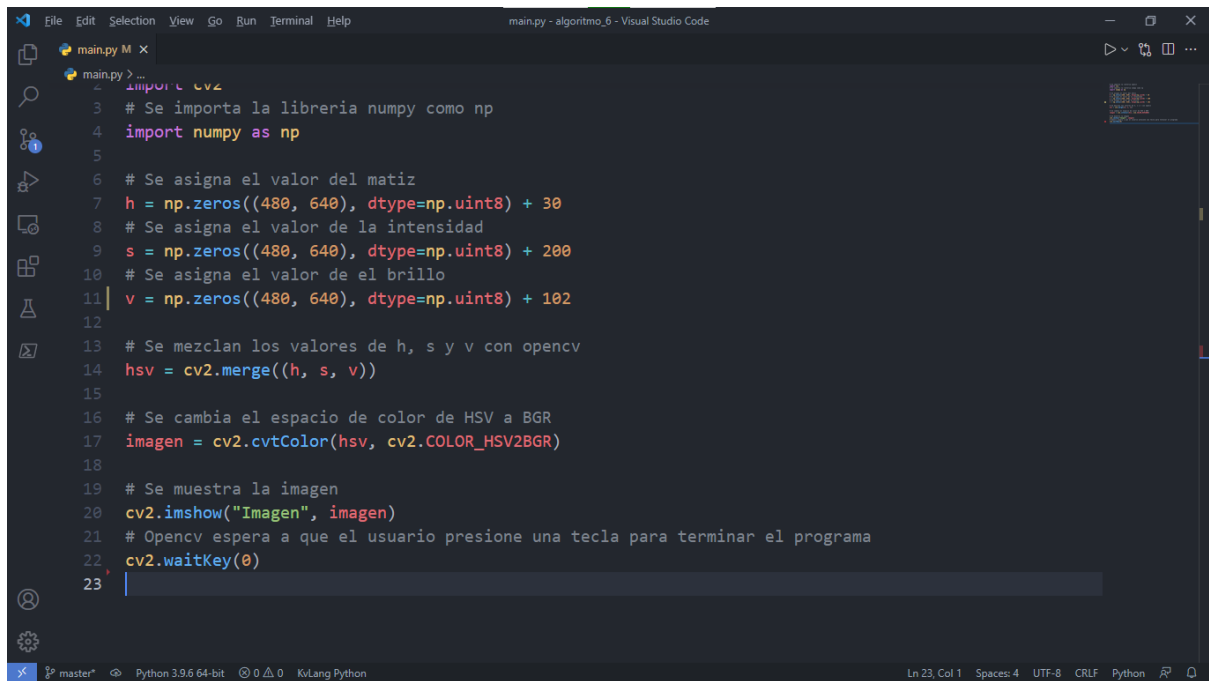
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 200
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 102

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow("Imagen", imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 4 de los ejercicios Value primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 200
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 102
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Figura 27: Código del punto 4

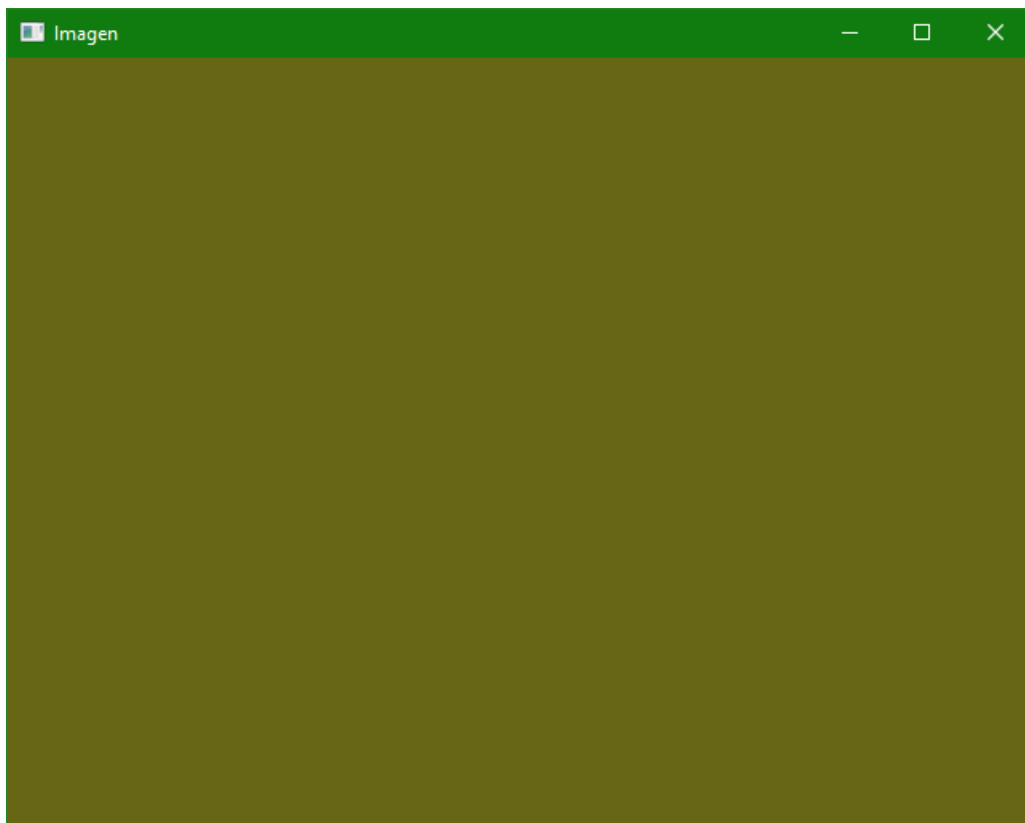


Figura 28: Imagen del punto 4

Código del punto 5 de los ejercicios Value

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

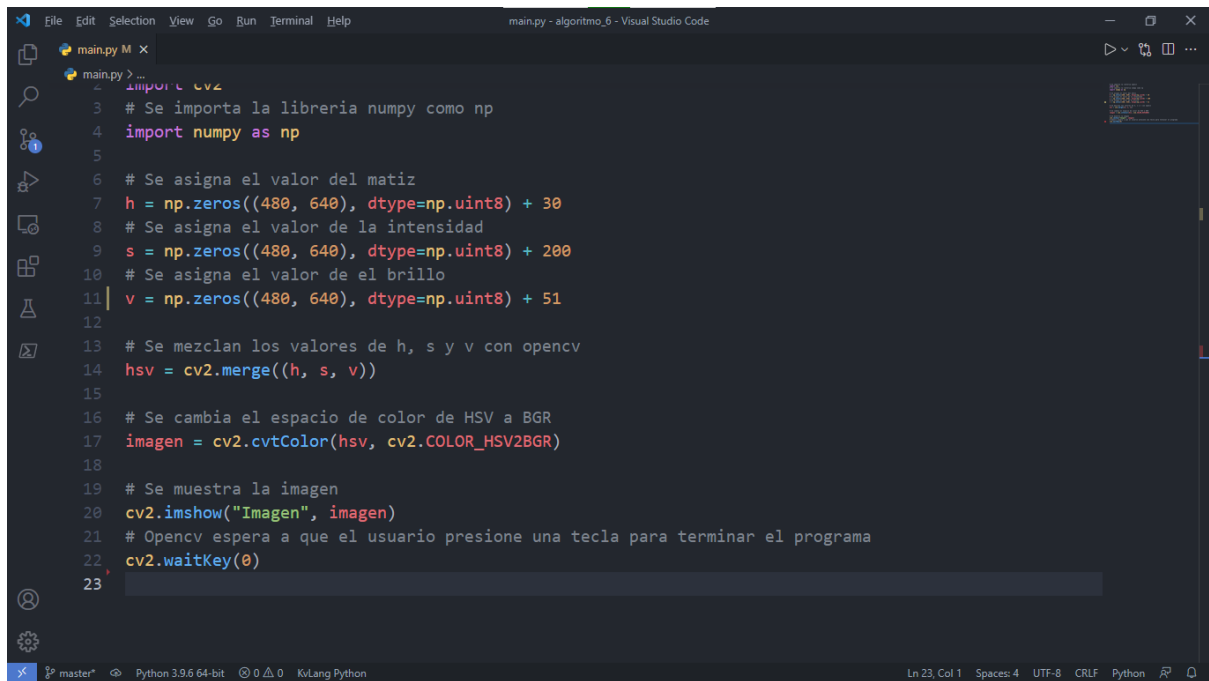
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 30
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 200
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 51

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código del punto 5 de los ejercicios Value primero se importan las librerías necesarias (*opencv*, *numpy*). Luego se asignan los valores de Hue a "h", Saturation a "s" y Value a "v", después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con *opencv*, al final *opencv* espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2 # Se importa la libreria numpy como np
3 import numpy as np
4
5 # Se asigna el valor del matiz
6 h = np.zeros((480, 640), dtype=np.uint8) + 30
7 # Se asigna el valor de la intensidad
8 s = np.zeros((480, 640), dtype=np.uint8) + 200
9 # Se asigna el valor de el brillo
10 v = np.zeros((480, 640), dtype=np.uint8) + 51
11
12 # Se mezclan los valores de h, s y v con opencv
13 hsv = cv2.merge((h, s, v))
14
15 # Se cambia el espacio de color de HSV a BGR
16 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
17
18 # Se muestra la imagen
19 cv2.imshow("Imagen", imagen)
20 # Opencv espera a que el usuario presione una tecla para terminar el programa
21 cv2.waitKey(0)
22
```

Figura 29: Código del punto 5

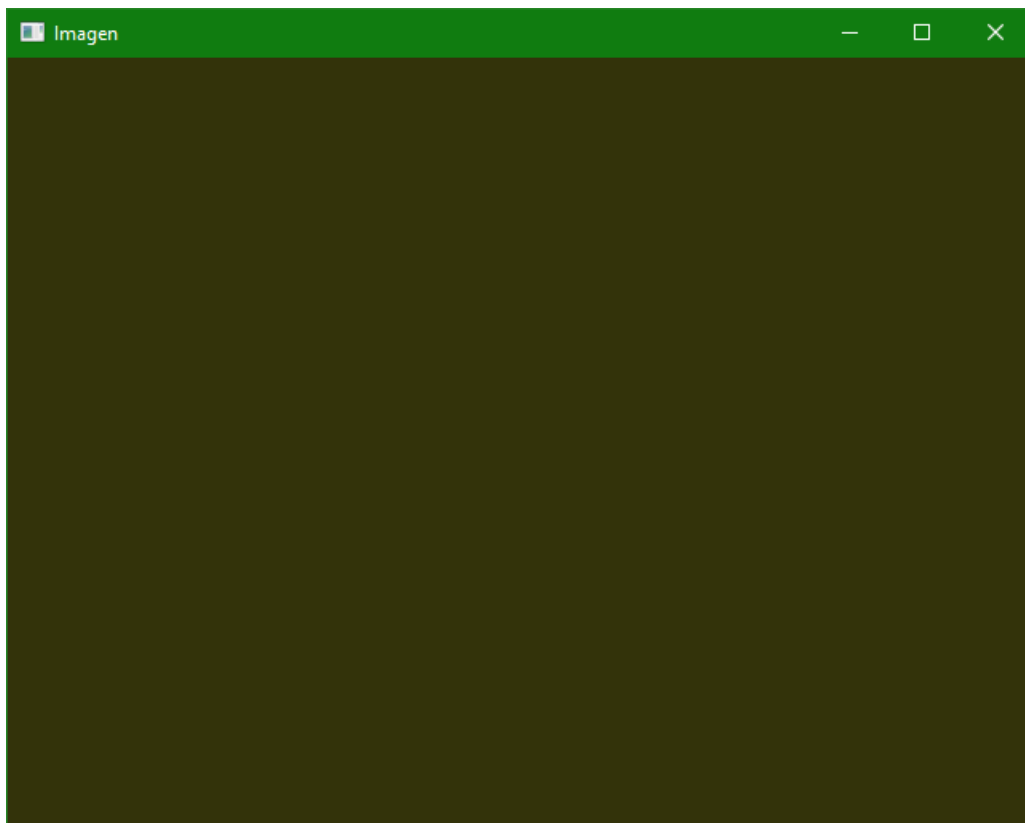


Figura 30: Imagen del punto 5

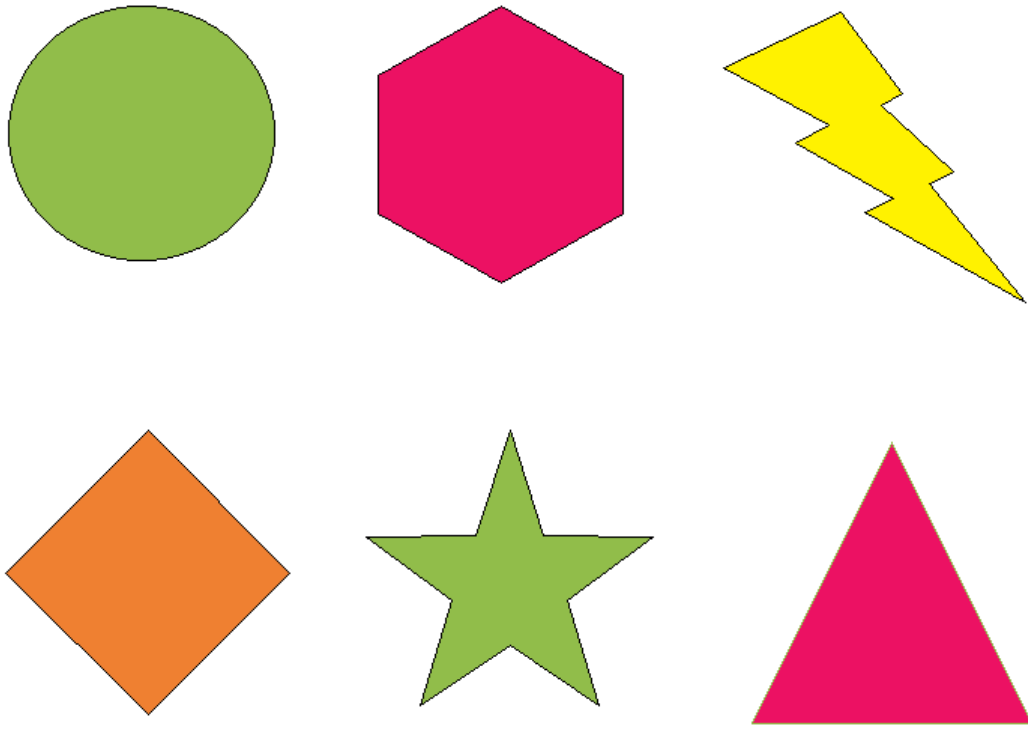


Figura 31: Figuras de color

Código de la imagen 1

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

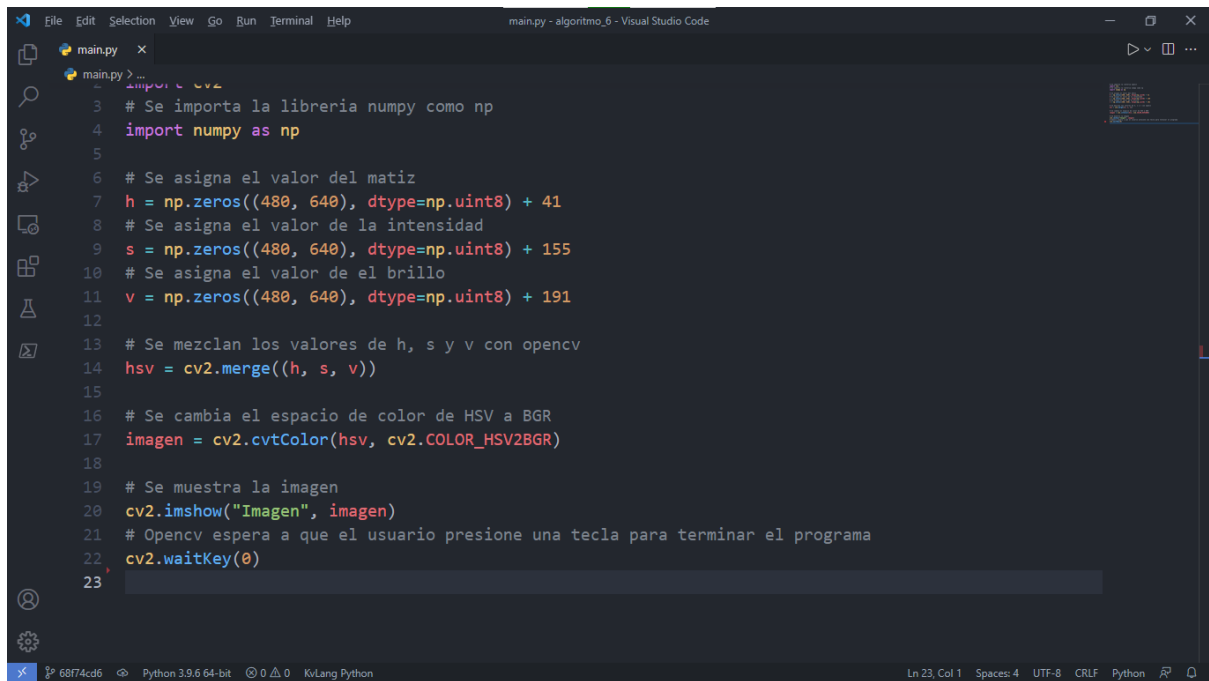
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 41
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 155
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 191

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código de la imagen 1, primero se importan las librerías necesarias (***opencv***, ***numpy***). Luego se asignan los valores de Hue, Saturation y Value en las variables "h", "s" y "v" respectivamente, después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con ***opencv***, al final ***opencv*** espera a que el usuario presione una tecla para terminar el programa.



```
1 import cv2
2
3 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 41
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 155
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 191
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 32: Código de la imagen 1

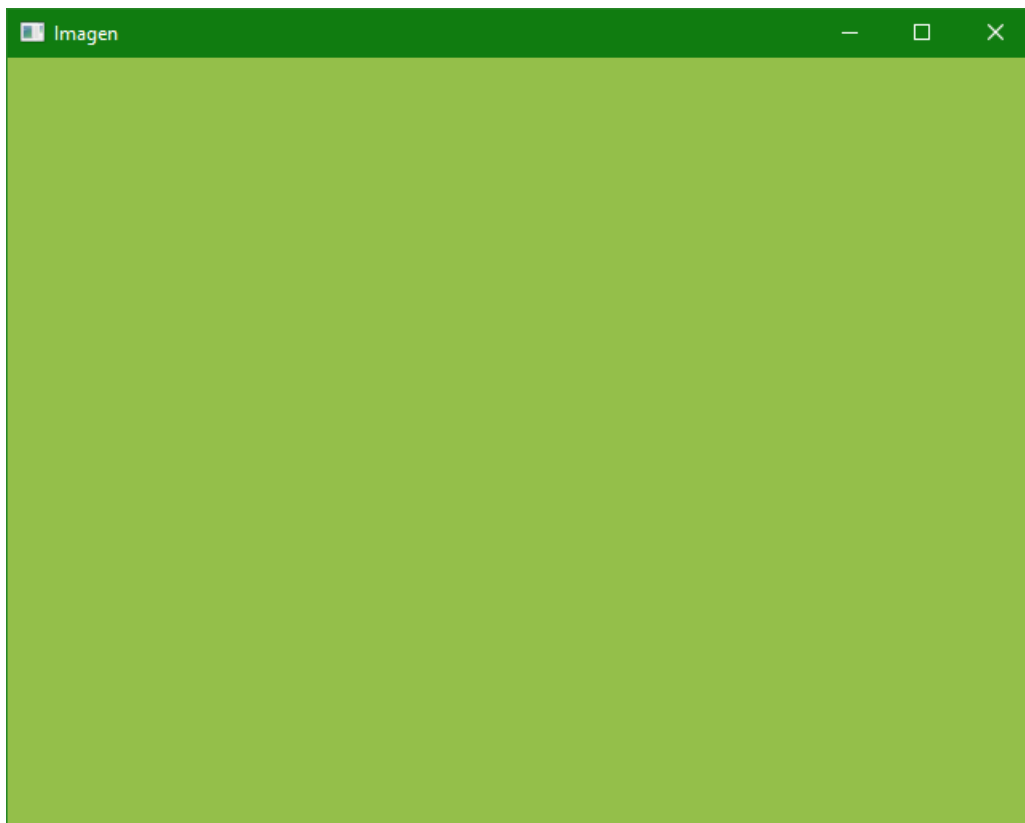


Figura 33: Imagen 1

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

# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 169
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 237
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 237

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código de la imagen 2, primero se importan las librerías necesarias (***opencv***, ***numpy***). Luego se asignan los valores de Hue, Saturation y Value en las variables "h", "s" y "v" respectivamente, después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con ***opencv***, al final ***opencv*** espera a que el usuario presione una tecla para terminar el programa.

```
File Edit Selection View Go Run Terminal Help
main.py - algoritmo_6 - Visual Studio Code

main.py M x
main.py > ...
1 # Se importa la libreria opencv
2 import cv2
3 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 169
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 237
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 237
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23 |
```

Figura 34: Código de la imagen 2

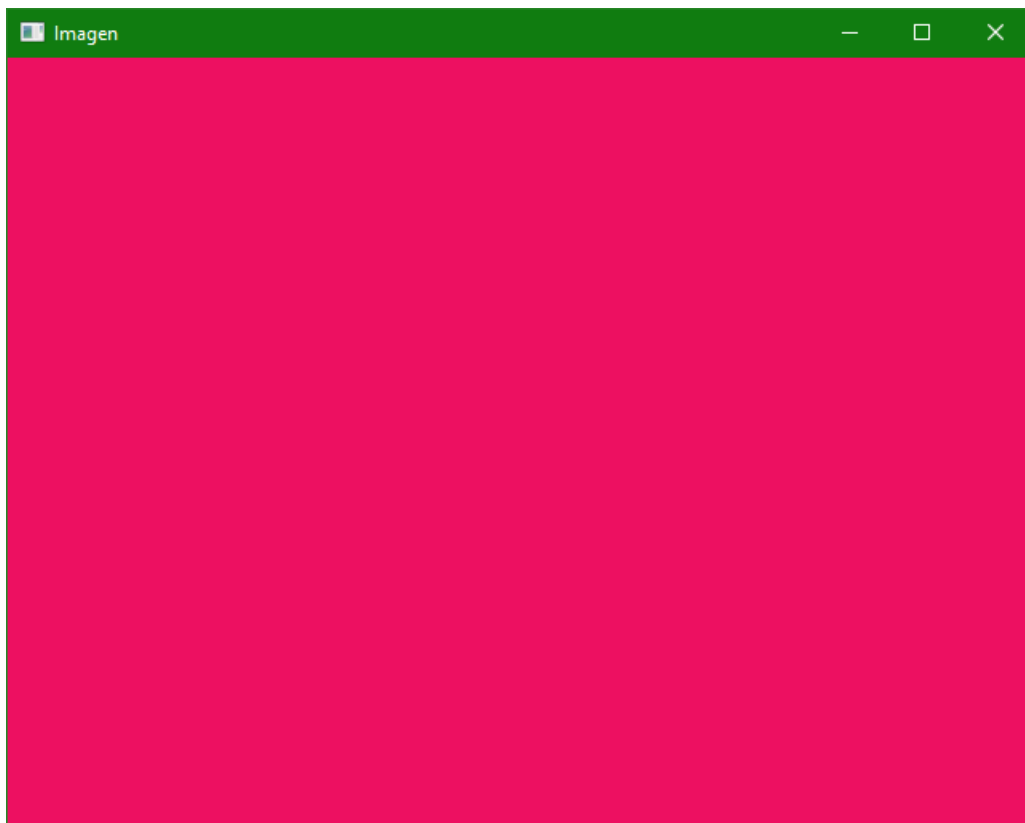


Figura 35: Imagen 2

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

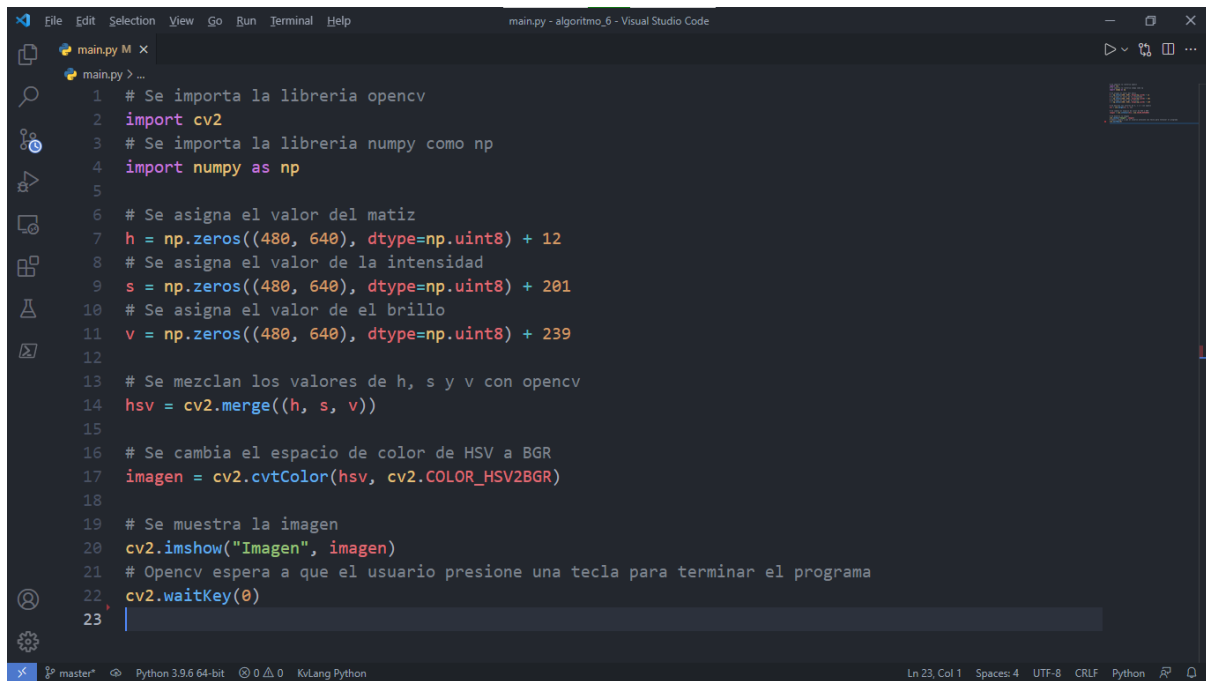
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 12
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 201
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 239

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código de la imagen 3, primero se importan las librerías necesarias (***opencv***, ***numpy***). Luego se asignan los valores de Hue, Saturation y Value en las variables "h", "s" y "v" respectivamente, después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con ***opencv***, 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 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 12
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 201
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 239
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 36: Código de la imagen 3

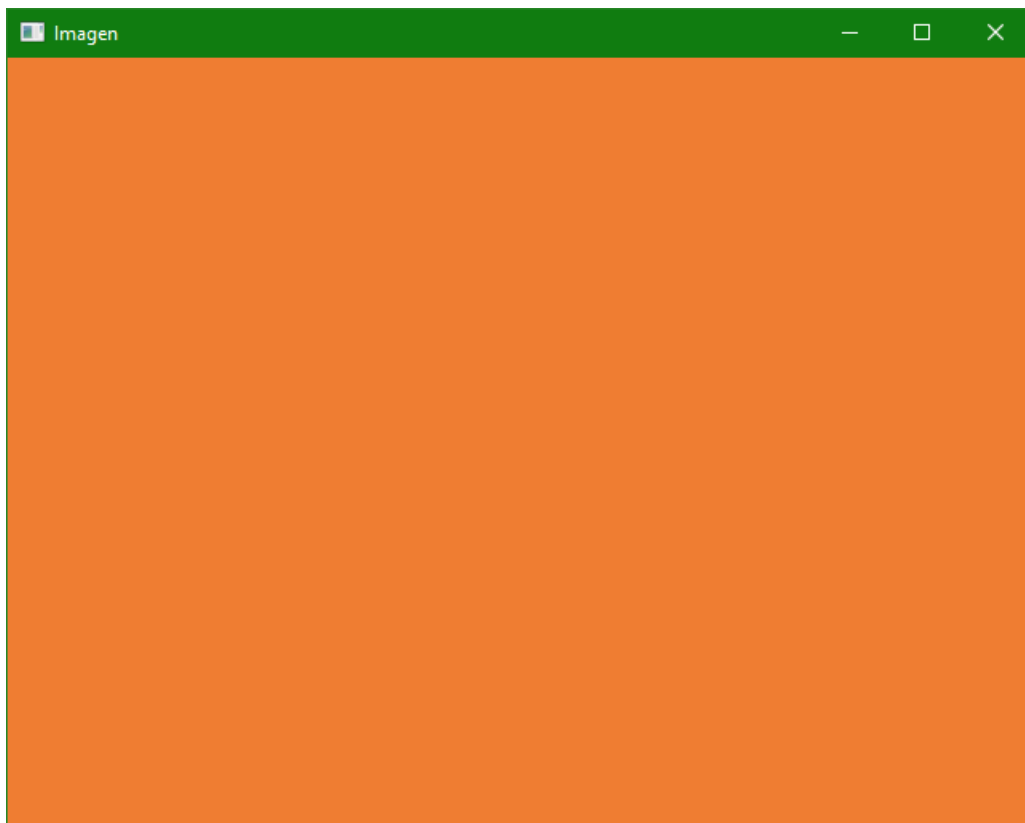


Figura 37: Imagen 3

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

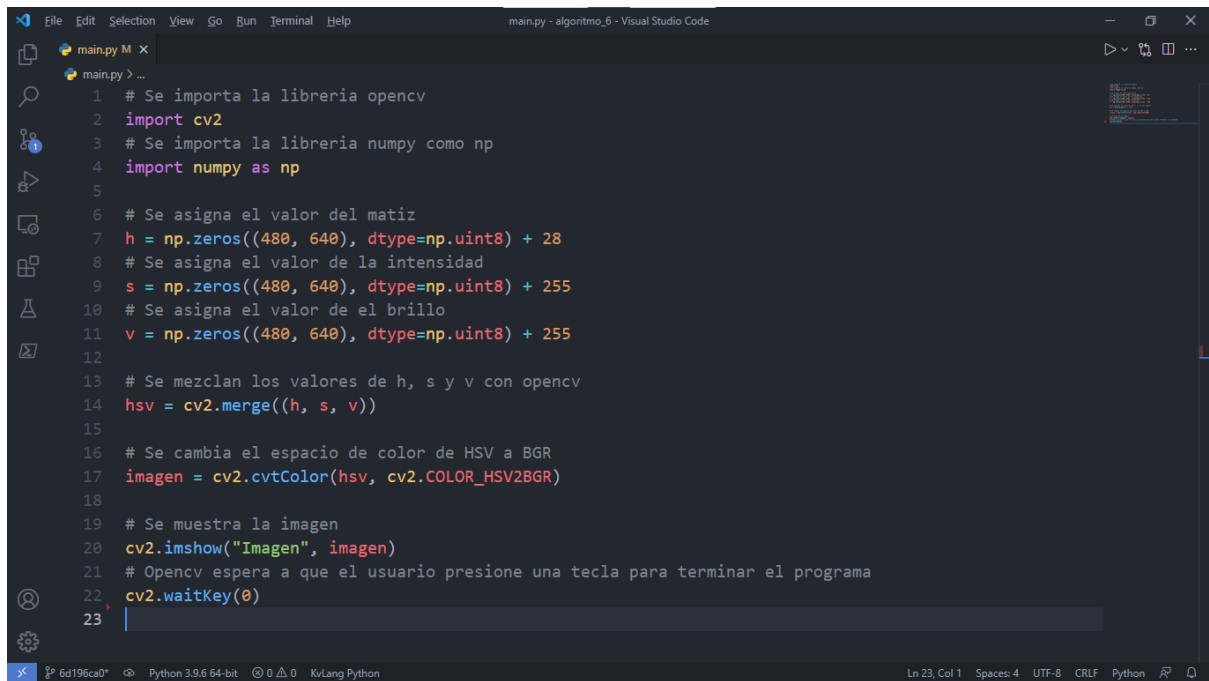
# Se asigna el valor del matiz
h = np.zeros((480, 640), dtype=np.uint8) + 28
# Se asigna el valor de la intensidad
s = np.zeros((480, 640), dtype=np.uint8) + 255
# Se asigna el valor de el brillo
v = np.zeros((480, 640), dtype=np.uint8) + 255

# Se mezclan los valores de h, s y v con opencv
hsv = cv2.merge((h, s, v))

# Se cambia el espacio de color de HSV a BGR
imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

# Se muestra la imagen
cv2.imshow('Imagen', imagen)
# Opcv espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el código de la imagen 4, primero se importan las librerías necesarias (***opencv***, ***numpy***). Luego se asignan los valores de Hue, Saturation y Value en las variables "h", "s" y "v" respectivamente, después se mezclan los valores en una sola imagen y esta se convierte de HSV a BGR. Por último se muestra la imagen con ***opencv***, 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 # Se importa la libreria numpy como np
4 import numpy as np
5
6 # Se asigna el valor del matiz
7 h = np.zeros((480, 640), dtype=np.uint8) + 28
8 # Se asigna el valor de la intensidad
9 s = np.zeros((480, 640), dtype=np.uint8) + 255
10 # Se asigna el valor de el brillo
11 v = np.zeros((480, 640), dtype=np.uint8) + 255
12
13 # Se mezclan los valores de h, s y v con opencv
14 hsv = cv2.merge((h, s, v))
15
16 # Se cambia el espacio de color de HSV a BGR
17 imagen = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
18
19 # Se muestra la imagen
20 cv2.imshow("Imagen", imagen)
21 # Opencv espera a que el usuario presione una tecla para terminar el programa
22 cv2.waitKey(0)
23
```

Figura 38: Código de la imagen 4

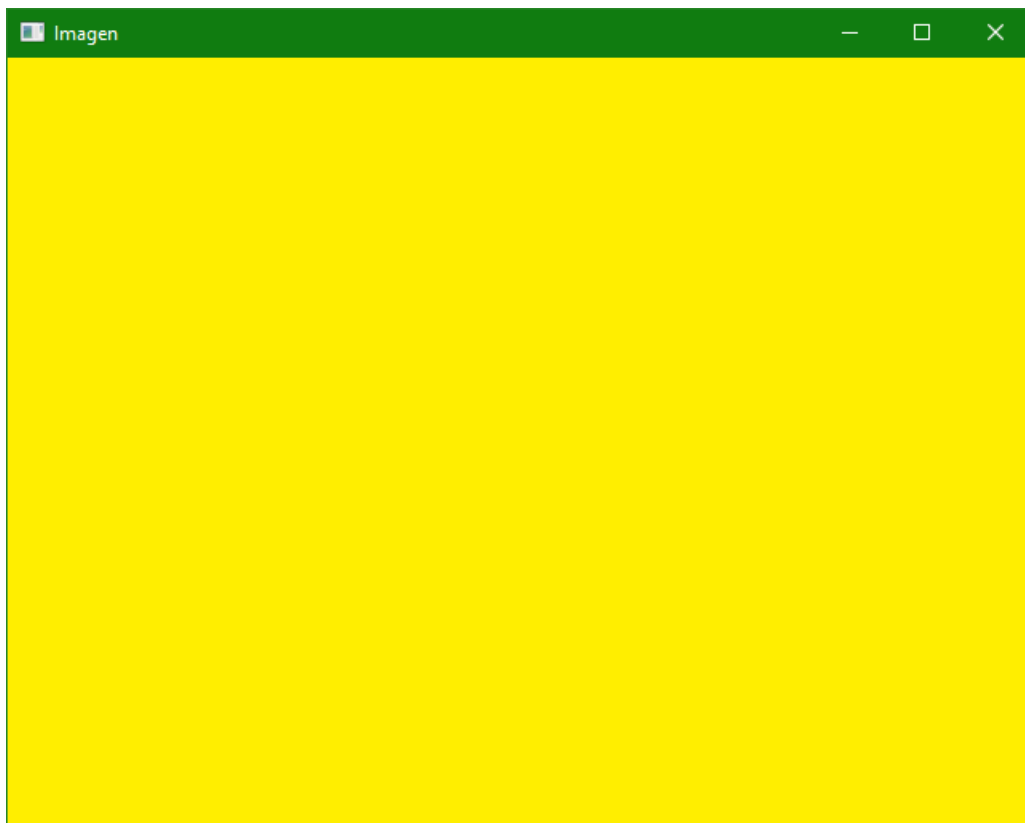


Figura 39: Imagen 4