

## Evaluación Continua N°2

Nombres y apellidos: Alexis Antonio Martinez Carrasquero

WI003952517

Fecha: 28/02/2023

1. Crear un programa en Python para obtener solamente la pelota de futbol a colores



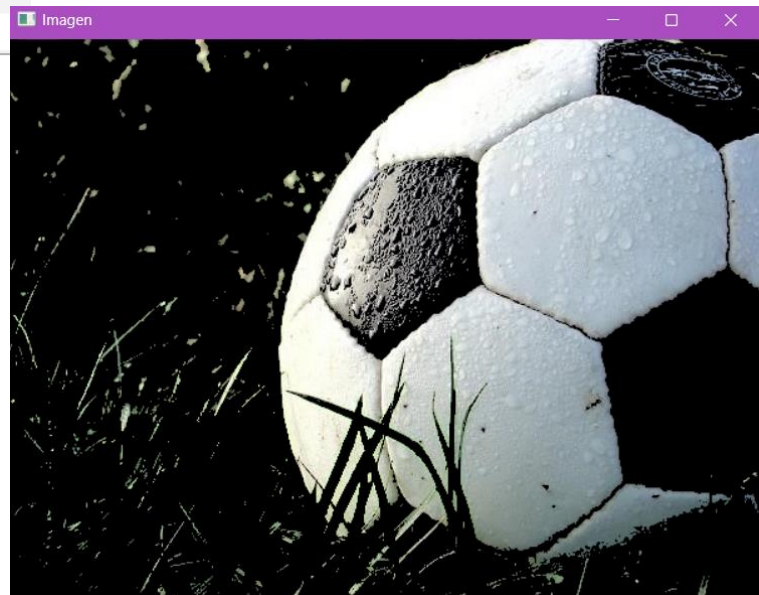
Ejercicio 01.py - C:/Users/1254/Desktop/PDI/Semana 07/EC2/Ejercicio 01.py

File Edit Format Run Options Window Help

#Alexis Martinez WI003952517

```
import cv2
import numpy as np

imagen = cv2.imread('Pelota_De_Futbol.jpg')
hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
lower = np.array([0, 0, 100])
upper = np.array([255, 70, 255])
mask = cv2.inRange(hsv, lower, upper)
pelota = cv2.bitwise_and(imagen, imagen, mask=mask)
cv2.imshow('Imagen', pelota)
```



2. Crear un programa en Python para obtener una imagen en binaria donde solo se pueda ver la pelota de tenis a colores



```
*Ejercicio 02, Pruebas.py - C:/Users/1254/Desktop/PDI/Semana 07/EC2/Ejercicio 02, Pruebas.py (3.10.11)*
File Edit Format Run Options Window Help
#Alexis Martinez WI003952517

import cv2
import numpy as np

imagen = cv2.imread('Tenis.jpg')
gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
_, binarizada = cv2.threshold(gris, 255, 255, cv2.THRESH_BINARY)
hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

lower = np.array([20, 50, 50])
upper = np.array([70, 255, 255])

mask = cv2.inRange(hsv, lower, upper)
pelota_color = cv2.bitwise_and(imagen, imagen, mask=mask)
resultado = cv2.bitwise_or(cv2.cvtColor(binarizada, cv2.COLOR_GRAY2BGR), pelota_color)

cv2.imshow('Tenis', resultado)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



3. Crear un programa en Python para obtener una imagen binaria donde se pueda ver las monedas. Considerar que al momento de binarizar la imagen puede aparecer ruido



\*Ejercicio 03.py - C:/Users/1254/Desktop/PDI/Semana 07/EC2/Ejercicio 03.py (3.10.11)\*

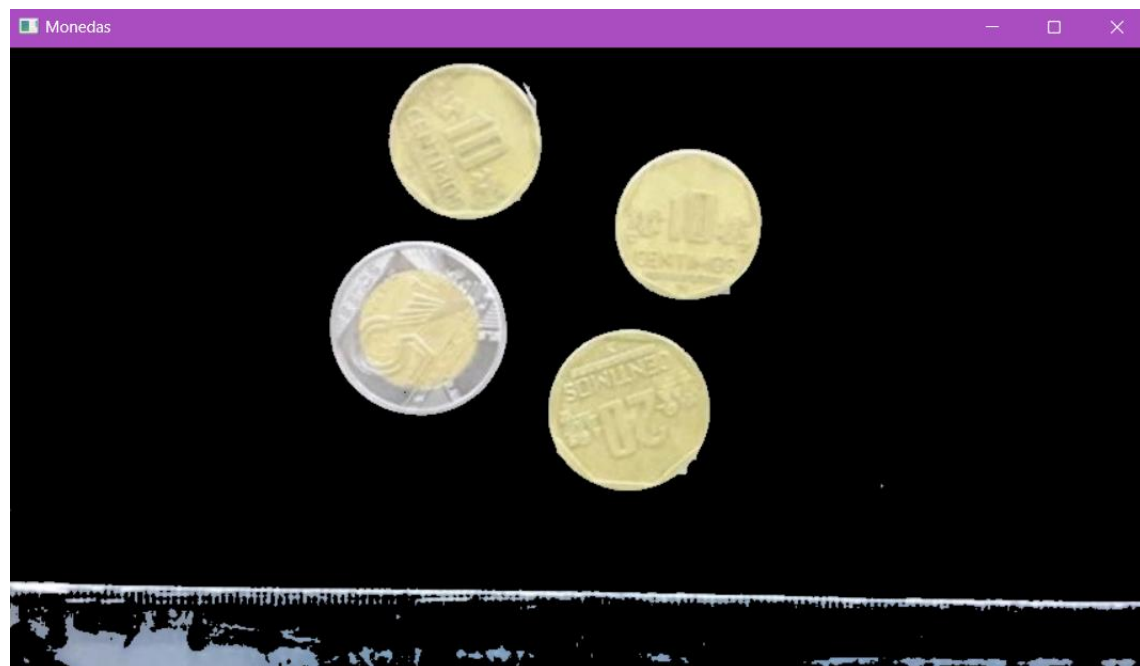
File Edit Format Run Options Window Help

#Alexis Martinez WI003952517

```
import cv2
import numpy as np

imagen = cv2.imread('Monedas.jpg')
gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
_, binarizada = cv2.threshold(gris, 143, 255, cv2.THRESH_BINARY)
resultado = cv2.bitwise_and(imagen, imagen, mask=binarizada)

cv2.imshow('Monedas', resultado)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



4. Crear un programa en Python para obtener una imagen en binario donde solo se pueda ver el limón a colores. Considerar que al momento de binarizar la imagen puede aparecer ruido



```
*Ejercicio 04.py - C:/Users/1254/Desktop/PDI/Semana 07/EC2/Ejercicio 04.py (3.10.11)*
File Edit Format Run Options Window Help
#Alexis Martinez WI003952517

import cv2
import numpy as np

imagen = cv2.imread('Limon.jpg')

gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
_, binarizada = cv2.threshold(gris, 255, 255, cv2.THRESH_BINARY)

binarizada_color = cv2.cvtColor(binarizada, cv2.COLOR_GRAY2BGR)

hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

lower = np.array([0, 80, 150])
upper = np.array([25, 250, 255])

mask = cv2.inRange(hsv, lower, upper)

pelota_color = cv2.bitwise_and(imagen, imagen, mask=mask)
resultado = cv2.bitwise_or(binarizada_color, pelota_color)

cv2.imshow('Limon', resultado)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

