

# Memoria de la practica 5 de VCO

Alumno:

-Vicente Burdeus Sánchez

## EJ1

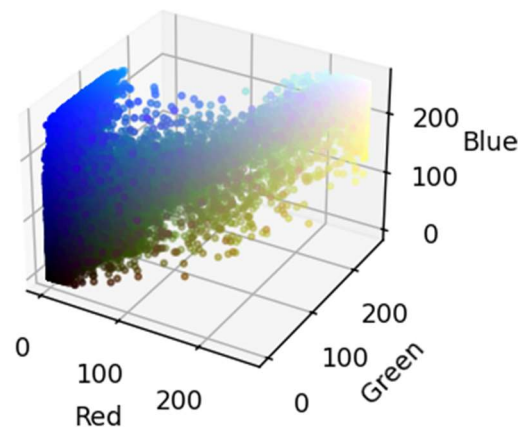
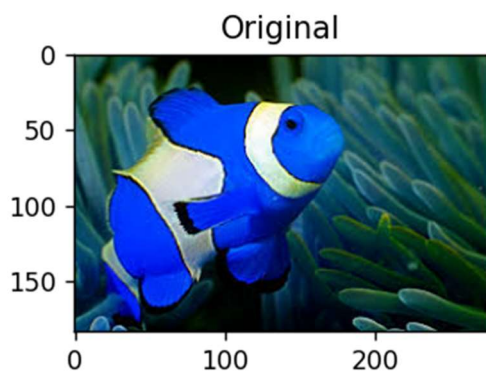
1. Escriba el código anterior y pruebe el resultado.

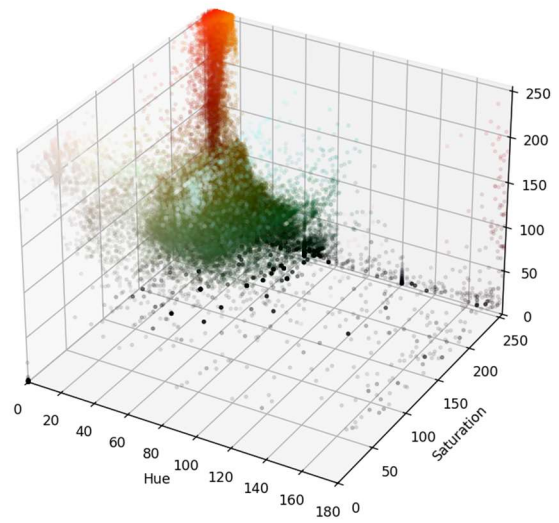
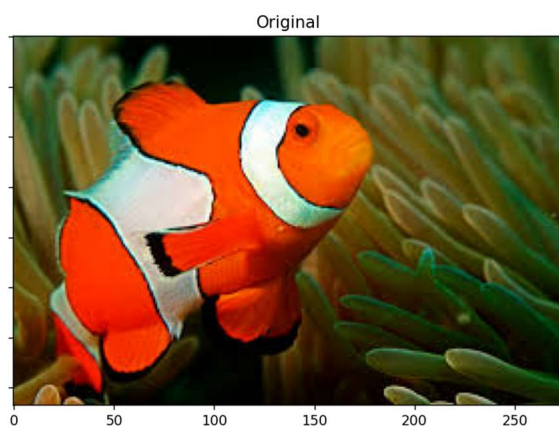
Supuestamente deberá obtener una imagen de nemo con tonos rojos, sin embargo, el pez se muestra en tonos azules.

¿Podría indicar a qué es debido esto e introducir la solución en el script?

2. Repita lo anterior, pero trabajando con la imagen en HSV. Deberá convertir a HSV y utilizar la función:

`h, s, v = cv2.split(nemo_hsv)`





## EJ2

1. Modifique el programa del ejercicio 1 para incluir la segmentación indicada en el código anterior.
2. Compruebe que funciona bien con las 6 imágenes de nemo que hay en la carpeta `./images/`.

Imagen Original



Máscara Naranja



Máscara Blanca



Máscara Combinada



Resultado Final

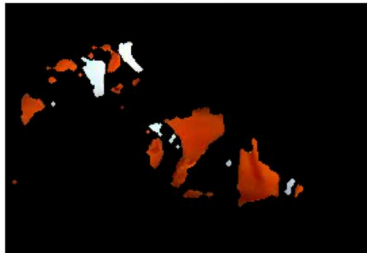


Imagen Original



Máscara Naranja



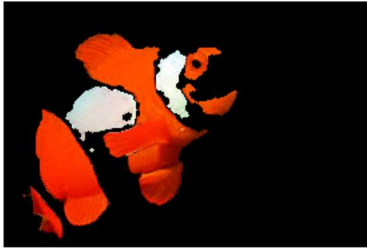
Máscara Blanca



Máscara Combinada



Resultado Final



Segmentación de nemo2.jpg

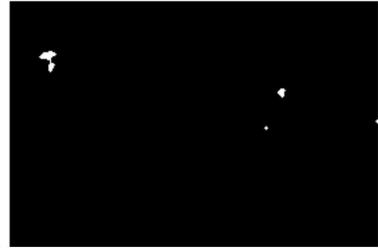
Imagen Original



Máscara Naranja



Máscara Blanca



Máscara Combinada



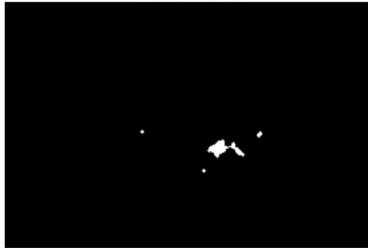
Resultado Final



Imagen Original



Máscara Naranja



Máscara Blanca



Máscara Combinada



Resultado Final



## EJ3

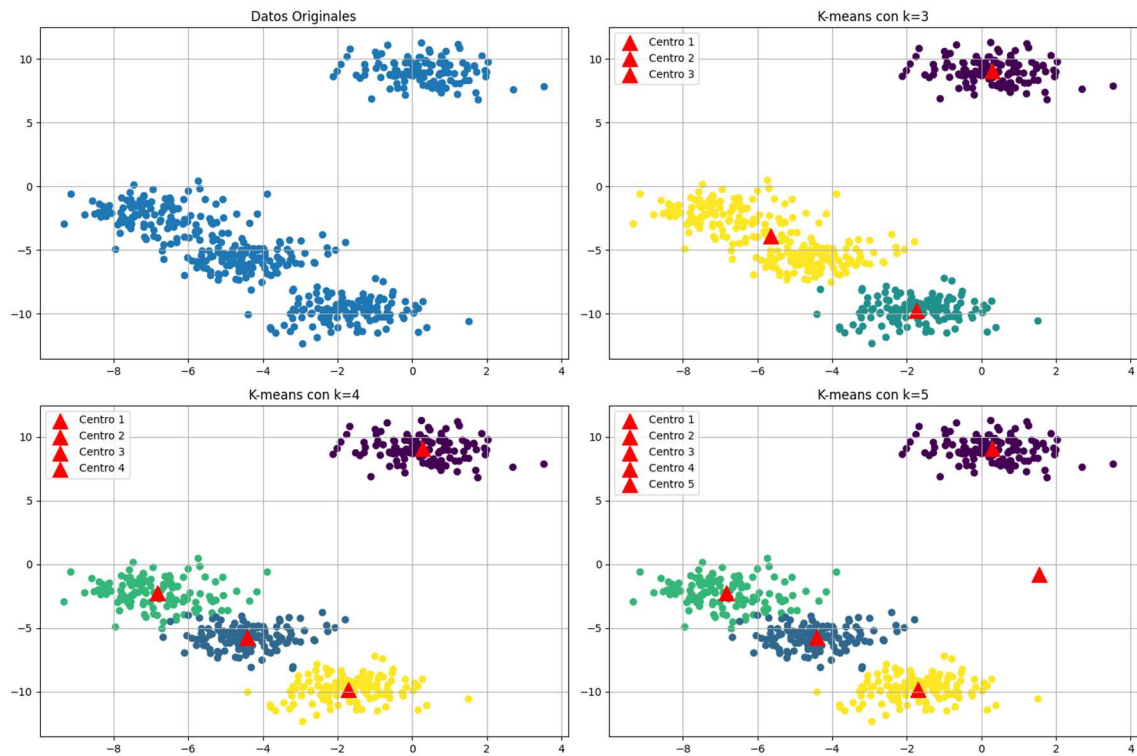
1. Modifique el programa del ejercicio 2 para añadir una segunda segmentación de los tonos blancos y obtenga una máscara `mask_white`
2. Combine ambas máscaras y muestre el resultado incluir la segmentación indicada en el código anterior.  
`final_mask = mask_orange + mask_white`  
`final_result = cv2.bitwise_and(nemo, nemo, mask=final_mask)`
3. Compruebe que funciona bien con las 6 imágenes de nemo que hay en la carpeta “./images/”.



2.

## Ej 4

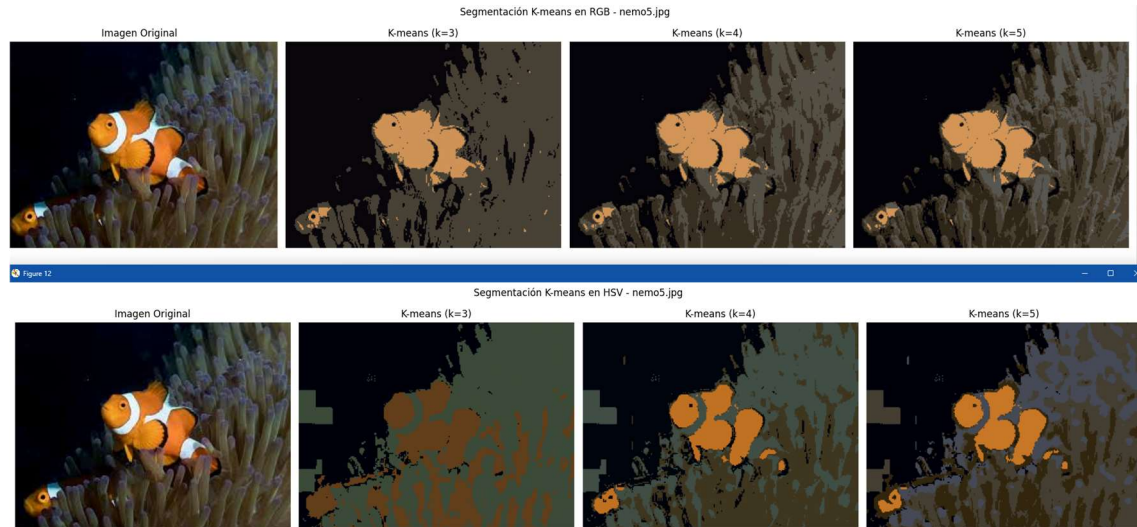
1. Tome el script 'k-means\_alt.py' disponible en PoliformaT .
2. Ejecute el código con k=3, 4 y 5 clases y compruebe el resultado



## EJ 5

1. Modifique el código anterior y cree el script "k-means-image.py" que permita hacer el algoritmo k\_Means con datos de tipo imagen. Ahora los datos serán de tamaño (n\_pixeles, 3) correspondientes a las 3 coordenadas de color (RGB, HSV, etc..).
2. En caso necesario, modifique la visualización en la función "plt.scatter()".

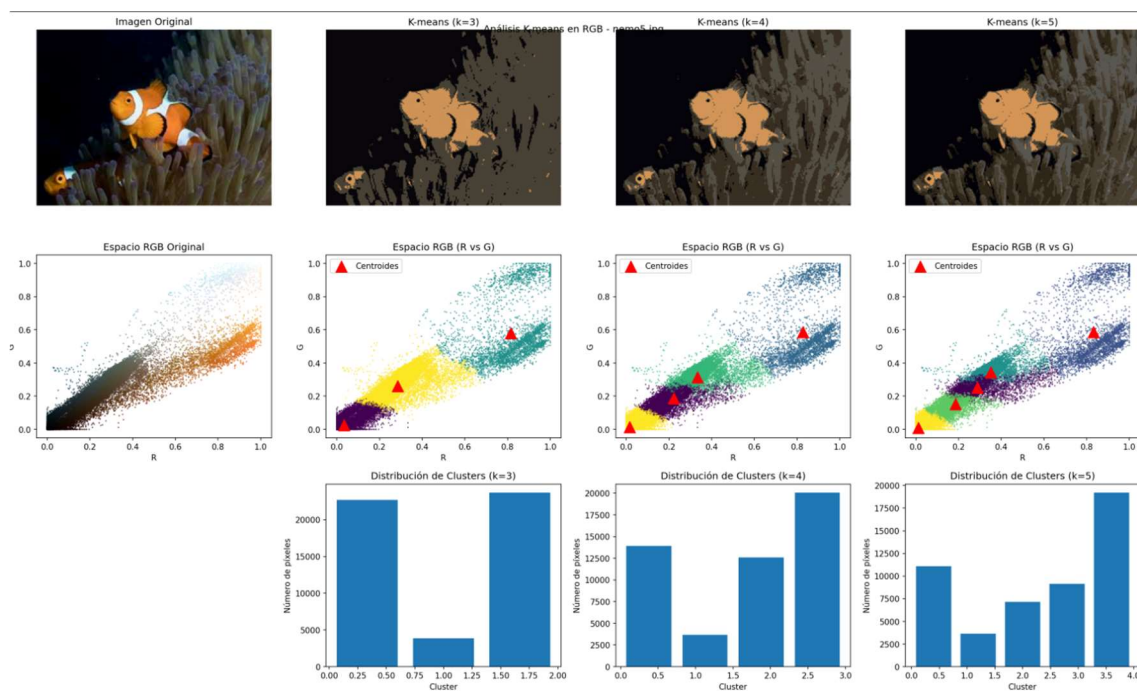
3. Utilice este algoritmo con las imágenes de nemo anteriormente indicadas.



*El resto de imágenes estas en la carpeta Practica 5 - > imágenes*

## EJ 6

1. Busque en internet el algoritmo k-Means++ y documente como sería
2. Busque un script Python de ese algoritmo y pruébelo.





## EJ 7

1. Copie el script anterior en un archivo 'MeanShift.py' y ejecútelo.
2. Compruebe el resultado.
3. Pruebe distintos valores del parámetro 'bandwidth' (20, 30, 40, ..).
4. Compruebe el resultado con otras imágenes de nemo.

