

```
In [1]: 1 !pip install imutils
```

Requirement already satisfied: imutils in /usr/local/lib/python3.7/dist-packages (0.5.4)

Importar paquetes y librerías

```
In [2]: 1 import matplotlib.pyplot as plt
2 import imutils
3 import cv2
```

Función para mostrar una imagen con imshow() de pyplot

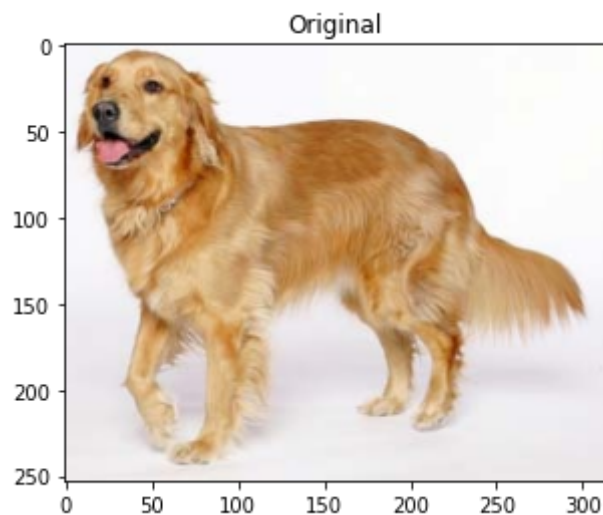
```
In [3]: 1 def mostrar_imagen(title, imagen):
2     imagen = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)
3     plt.imshow(imagen)
4     plt.title(title)
5     plt.grid(False)
6     plt.show()
```

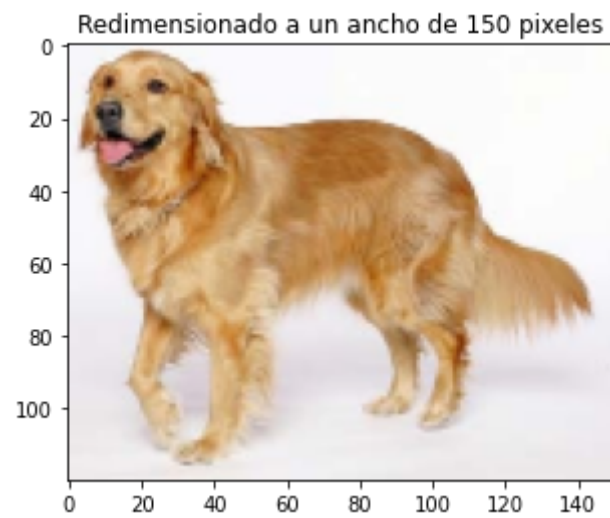
Implementación de redimensionamiento básico con OpenCV

```
In [4]: 1 args = {
2     "imagen" : "labrador.jpg"
3 }
```

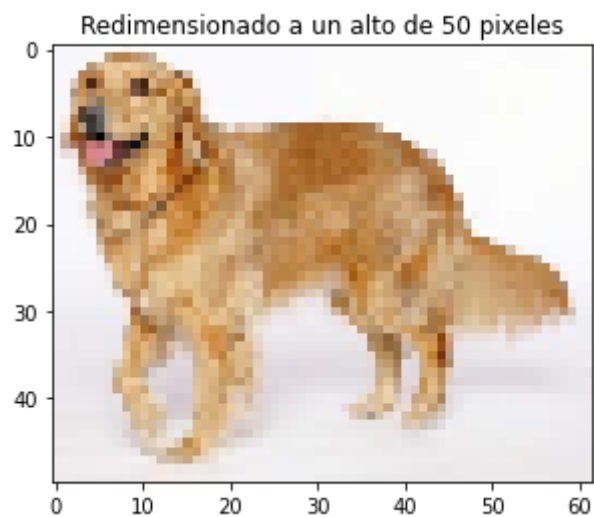
In [10]:

```
1  #Cargar la imagen y mostrarla
2  imagen = cv2.imread(args["imagen"])
3  mostrar_imagen("Original",imagen)
4
5  # Redimensionar la imagen para tener 150 px de ancho
6  # haciendo que la otra dimensión tenga un tamaño
7  # apropiado y no se distorsione
8
9  # proporción de reducción de la imagen
10 r = 150.0/imagen.shape[1]
11
12 # nueva dimensión de la imagen
13 dim = (150, int(imagen.shape[0]*r))
14
15 # redimensionar la imagen
16 redimensionado = cv2.resize(imagen,dim,cv2.INTER_AREA)
17 mostrar_imagen("Redimensionado a un ancho de 150 pixeles",redimensionado)
```

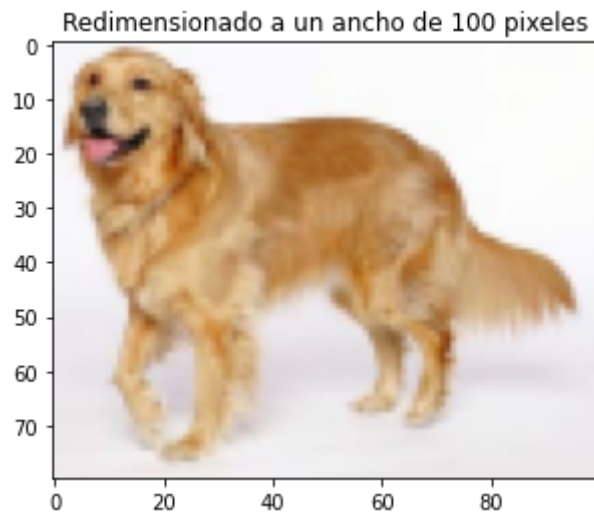




```
In [11]: 1 # redimensionamiento de la imagen a un alto de 50 pixeles
2 # manteniendo la proporción de la imagen
3
4 r = 50.0/imagen.shape[0]
5
6 # nueva dimensión de la imagen
7 dim = (int(imagen.shape[1]*r),50)
8
9 # redimensionar la imagen
10 redimensionado = cv2.resize(imagen,dim,cv2.INTER_AREA)
11 mostrar_imagen("Redimensionado a un alto de 50 pixeles",redimensionado)
```



```
In [13]: 1 # redimensionando con imutils a un ancho de 100 pixeles
2 redimensionado = imutils.resize(imagen,width=100)
3 mostrar_imagen("Redimensionado a un ancho de 100 pixeles",redimensionado)
```

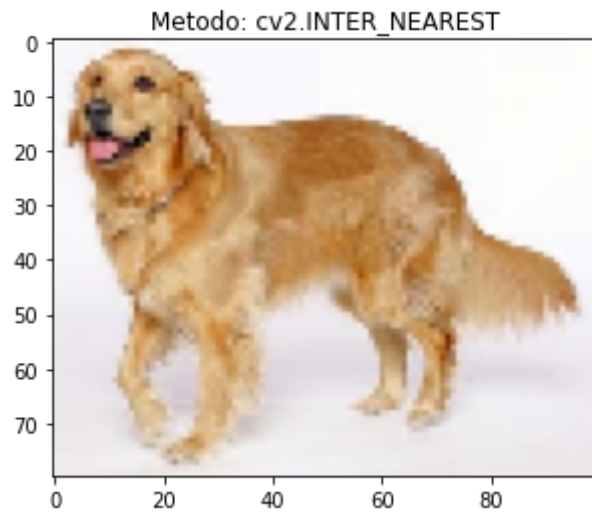


```
In [ ]: 1 Comparación de métodos de interpolación con OpenCV
```

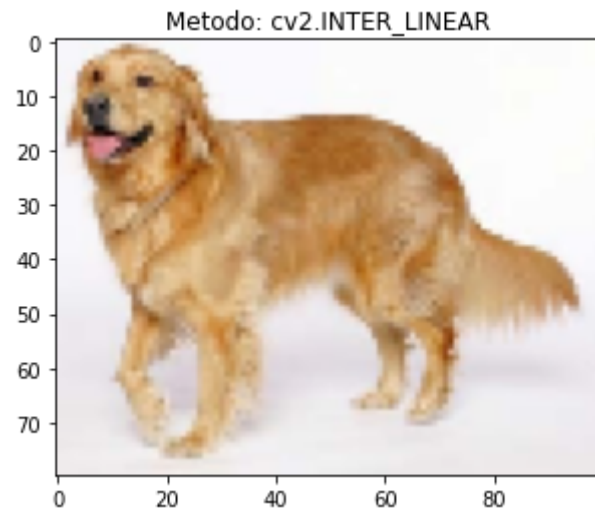
```
In [14]: 1 metodos = [("cv2.INTER_NEAREST",cv2.INTER_NEAREST),
2             ("cv2.INTER_LINEAR",cv2.INTER_LINEAR),
3             ("cv2.INTER_AREA",cv2.INTER_AREA),
4             ("cv2.INTER_CUBIC",cv2.INTER_CUBIC),
5             ("cv2.INTER_LANCZOS4",cv2.INTER_LANCZOS4)]
```

```
In [15]: 1 for (nombre,metodo) in metodos:
2         print("Método:{}".format(nombre))
3         redimensionado = imutils.resize(imagen,width=100,inter=metodo)
4         mostrar_imagen("Método: {}".format(nombre),redimensionado)
```

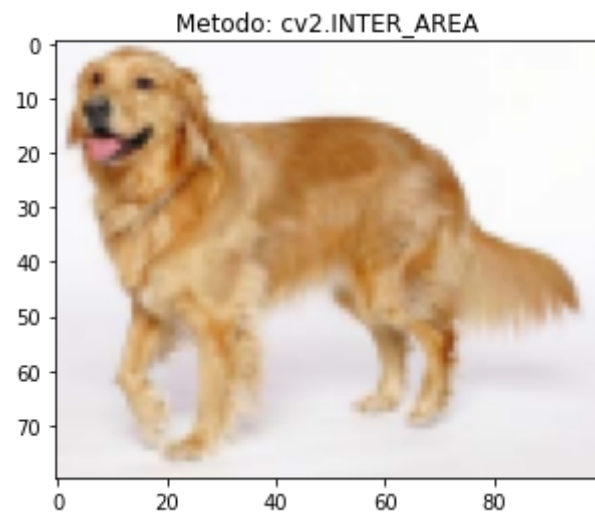
Método:cv2.INTER\_NEAREST



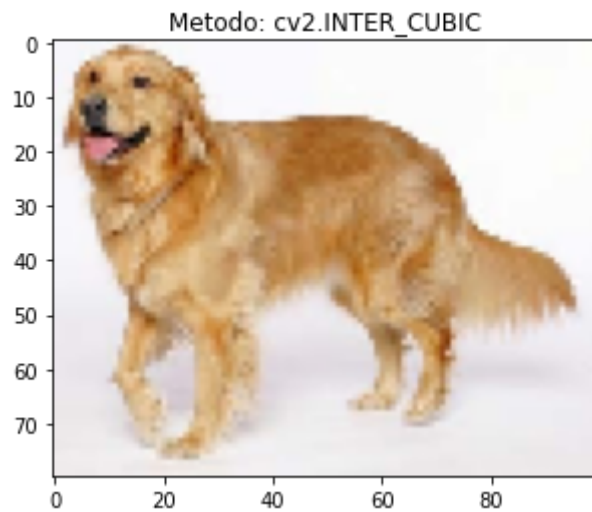
Método:cv2.INTER\_LINEAR



Método:cv2.INTER\_AREA



Método: `cv2.INTER_CUBIC`



Método: `cv2.INTER_LANCZOS4`

