

# Trabajo\_Alberto\_Isorna

April 24, 2019

## 1 Trabajo: Familiarizándonos con el entorno de procesamiento de imágenes

Elaboración de un notebook básico de python donde se carguen los módulos de numpy y scikit image

```
In [1]: from skimage import data
        from skimage import io
        from skimage.transform import resize
        from skimage import color

        import matplotlib.pyplot as plt
        import numpy as np
```

Cargar una imagen de su disco duro

```
In [2]: myim = io.imread("Paisaje.jpg")
```

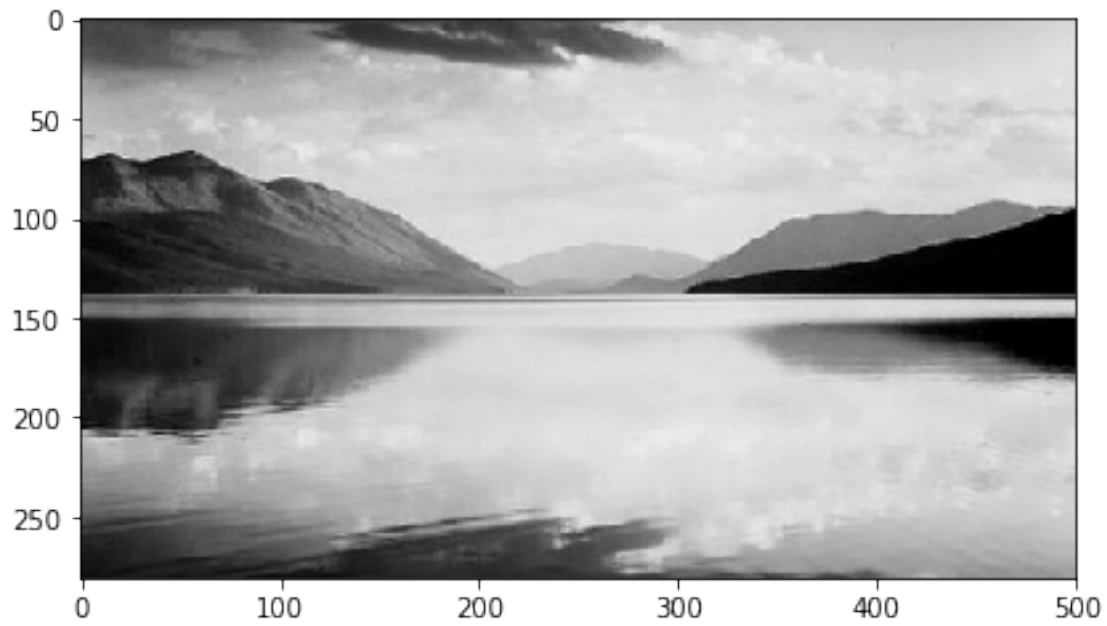
Obtener la distribución y su representación de la suma de las componentes de la imagen de forma vertical y horizontal

```
In [3]: # la imagen tiene 3 canales asi que hacemos un cambio a gris
        myimg = color.rgb2gray(myim)
        print('shape = {}'.format(myimg.shape))

        io.imshow(myimg)
```

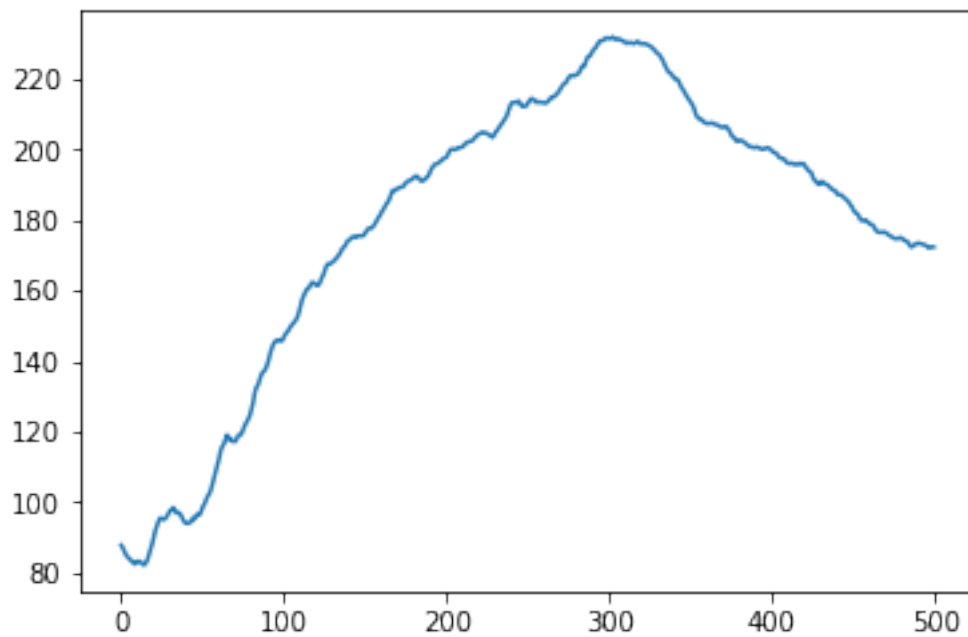
```
shape = (281, 500)
```

```
Out[3]: <matplotlib.image.AxesImage at 0x1e4713f2da0>
```



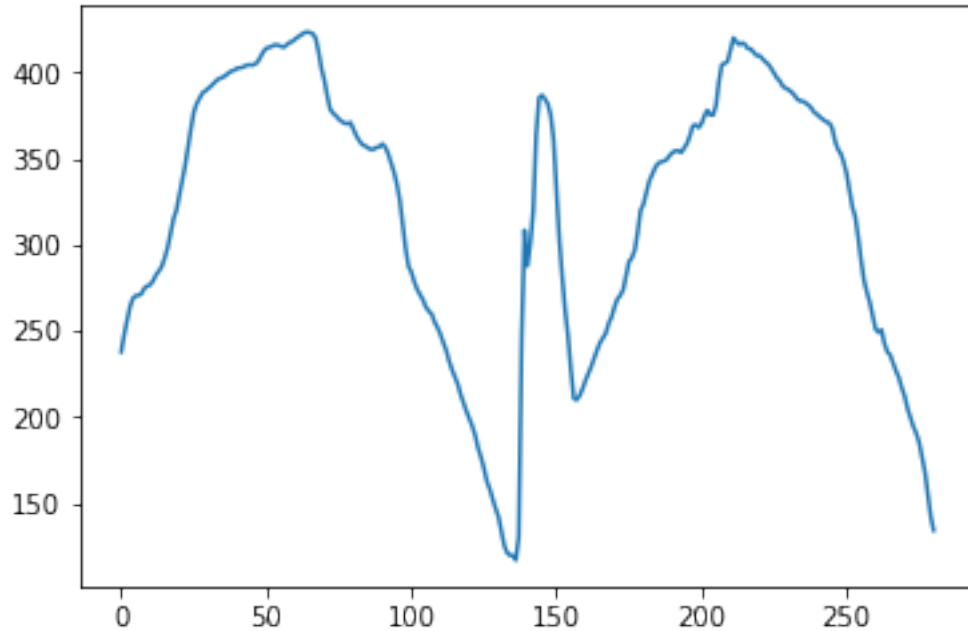
```
In [4]: # obtenemos la suma de vectores
        suma_c = np.sum(myimg, axis=0) #columnas axis = 0
        suma_f = np.sum(myimg, axis=1) #filas axis = 1

In [5]: # Histograma por columnas
        plt.plot(suma_c)
        plt.show()
```



Si analizamos por columnas, vemos como por terminos generales la imagen es mas clara en el centro

```
In [6]: # Histograma por filas
plt.plot(suma_f)
plt.show()
```



Si analizamos por filas, como si leyeramos la imagen de arriba a abajo, primero empieza siendo

Obtener el valor máximo y mínimo de la imagen y su posición

```
In [7]: # MAXIMO
xmax = np.max(myimg)
print(np.argmax(myimg)) #esto es posicion contando de uno en uno

ind = np.unravel_index(np.argmax(myimg, axis=None), myimg.shape)
print('Max = {} (escala de 0 a 1)'.format(xmax))
print('Posicion (Y,X) max : {}'.format(ind))
```

34505

Max = 1.0 (escala de 0 a 1)

Posicion (Y,X) max : (69, 5)

```
In [8]: # MINIMO
        xmin = np.min(myimg)

        ind2 = np.unravel_index(np.argmin(myimg, axis=None), myimg.shape)
        print('Min = {} (escala de 0 a 1)'.format(xmin))
        print('Posicion (Y,X) min : {}'.format(ind))
```

```
Min = 0.0 (escala de 0 a 1)
Posicion (Y,X) min : (69, 5)
```

Para cada valor calculado anteriormente, pintar dentro de la imagen una cruz de 5 píxeles x 5 píxeles.

```
In [9]: myimg_2 = myimg
        d = 2 #radio

        cx = ind[1]
        cy = ind[0]

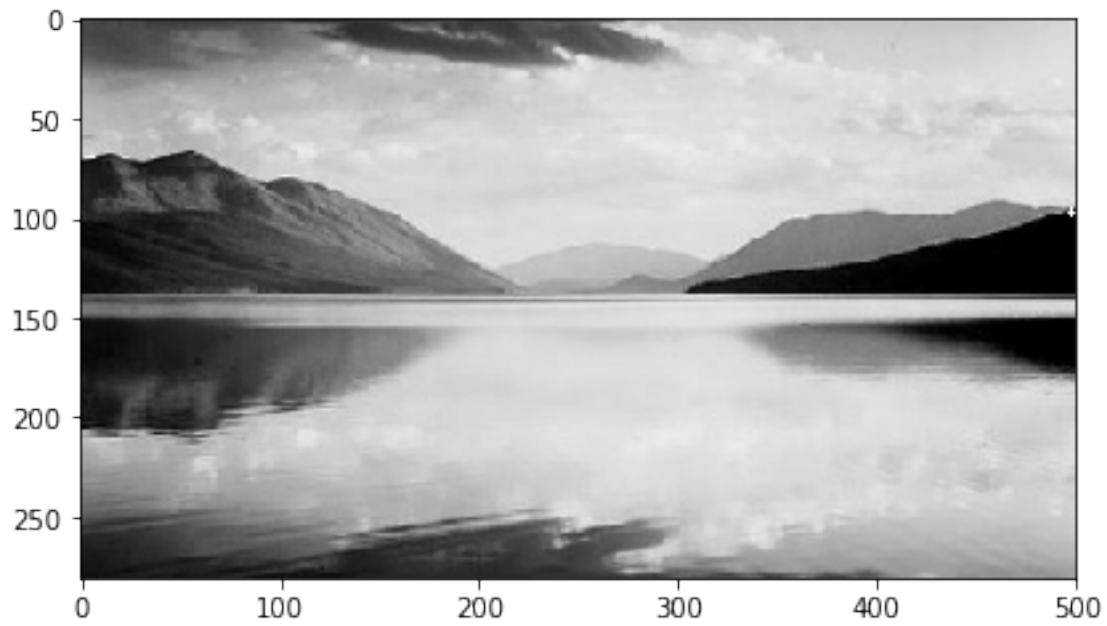
        # Pintamos maximo
        myimg_2[(cy-d):(cy+d+1),cx]= 1
        myimg_2[cy,(cx-d):(cx+d+1)] = 1
        # no se aprecia porque esta escondida en el margen izquierdo

        cx = ind2[1]
        cy = ind2[0]

        # Pintamos minimo
        myimg_2[(cy-d):(cy+d+1),cx]= 1
        myimg_2[cy,(cx-d):(cx+d+1)] = 1
        #a la derecha de la imagen
```

```
In [10]: io.imshow(myimg_2)
```

```
Out[10]: <matplotlib.image.AxesImage at 0x1e4713bd160>
```



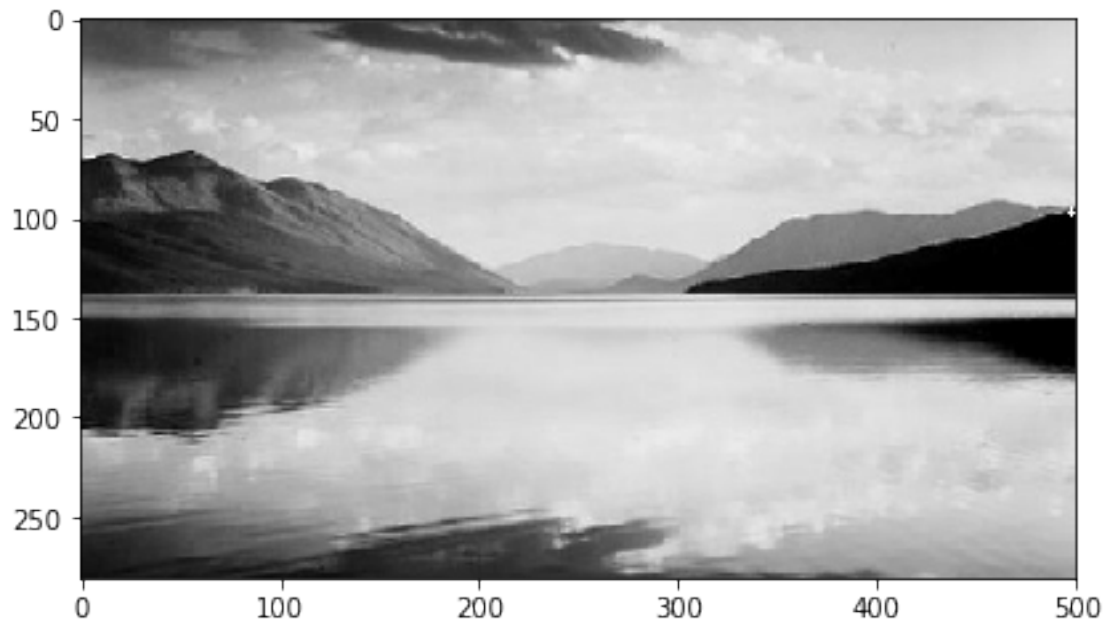
```
In [11]: myimg_2 = np.arange(25).reshape(5,5)
         cx = 2
         cy = 2
         d = 1
```

```
In [12]: cy
```

```
Out[12]: 2
```

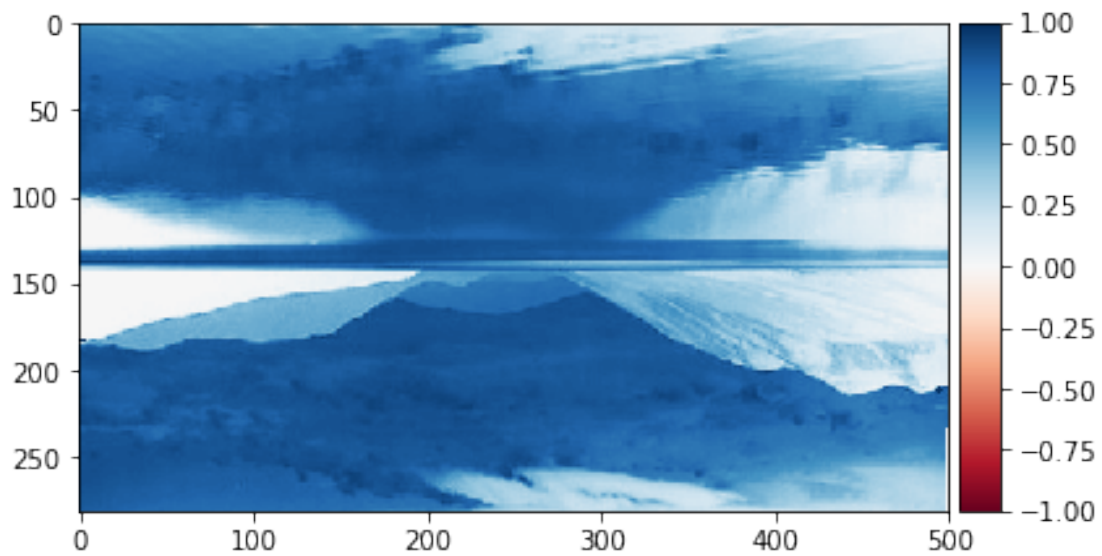
```
In [13]: from scipy import ndimage
         img2=myimg
         io.imshow(img2)
```

```
Out[13]: <matplotlib.image.AxesImage at 0x1e4728b0fd0>
```



```
In [14]: rotate_img2 = ndimage.rotate(img2, 180)
         io.imshow(rotate_img2)
```

```
Out[14]: <matplotlib.image.AxesImage at 0x1e472b56ba8>
```



```
In [ ]:
```