

# **Procesamiento Digital de Imágenes**

Christian Hernández,  
Noviembre 2020.

Universidad Politécnica Salesiana.  
Ingeniería de Sistemas.  
Intelligence Artificial.

In recent years digital processing of images has gained an important place in the field of information technology and computing. Presently, it is the base of a growing variety of applications which includes: medical diagnosis, remote perception, space exploration, computer vision, etc. As a result of the reduction of computers prices, presently, the digital processing of images can be accomplished with a personal computer. The present paper offers a brief introduction to this area of computing. It refers to the main theories and methods, and shows the results of these theories and methods when used with certain images

## Capítulo 1

### Definición

El procesamiento digital de imágenes ha adquirido, en años recientes, un papel importante en las tecnologías de la información y el cómputo. Actualmente, es la base de una creciente variedad de aplicaciones que incluyen diagnóstico médica, percepción remota, exploración espacial, visión por computadora, etc. Como resultado directo de la reducción en el precio de las computadoras, el procesamiento digital de imágenes actualmente se puede efectuar en una computadora personal. El presente trabajo proporciona una breve introducción a esta área de la informática y de la computación haciendo referencia a las principales teorías y métodos; asimismo, se muestran los resultados de estas teorías y métodos cuando se aplican a imágenes dadas

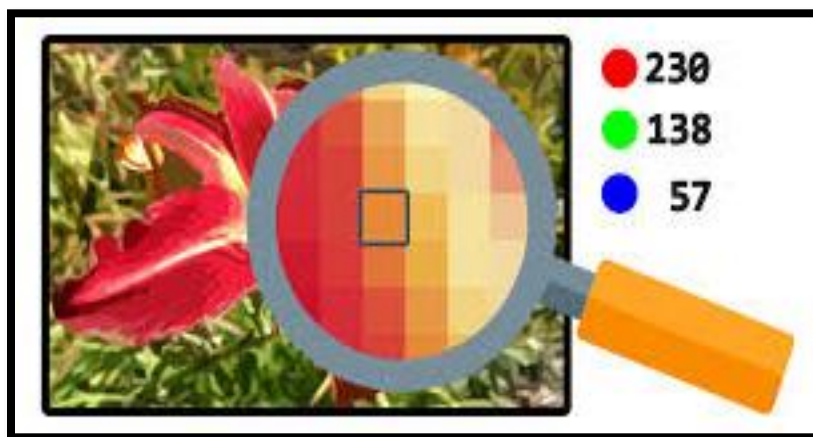


Figura1: Procesamiento de Imágenes

## Procesamiento Digital de Imágenes

Es todo un conjunto de técnicas que tiene como objetivos, obtener una representación más adecuada de una imagen, o bien, es también utilizado en la extracción de características relevantes de esta, las cuales muchas veces no son perceptibles a primera vista

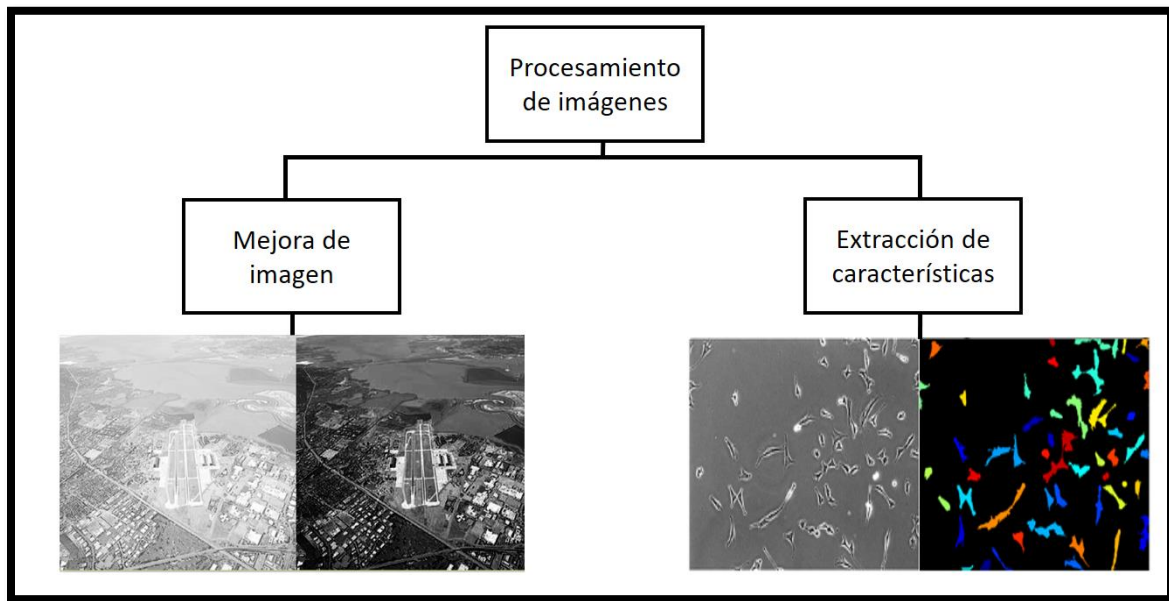


Figura2: Imágenes

### Ventajas

- Método simple y sencillo de implementar.
- Fácil asociación del concepto de frecuencia con ciertas características de la imagen; cambios de tonalidad suaves implican frecuencias bajas y cambios bruscos frecuencias altas.
- Proporciona flexibilidad en el diseño de soluciones de filtrado.
- Rapidez en el filtrado al utilizar el Teorema de la Convolución.

### Desventajas

- Se necesitan conocimientos en varios campos para desarrollar una aplicación para el procesamiento de imágenes.
- El ruido no puede ser eliminado completamente.

## Ejemplos de Aplicación de PDI

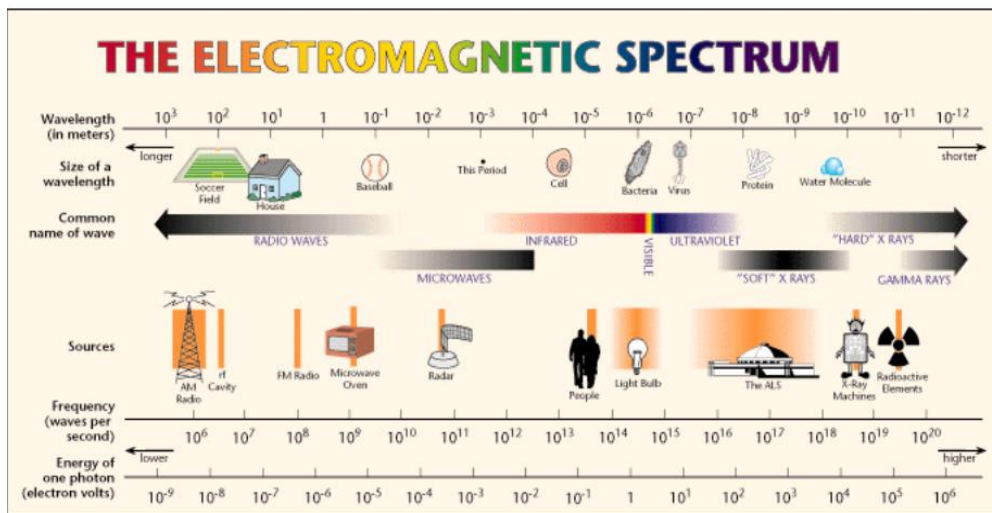
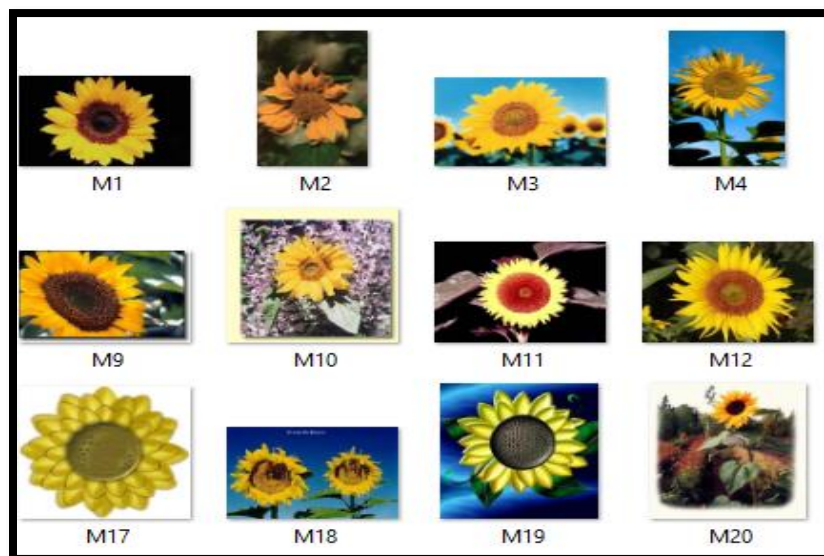


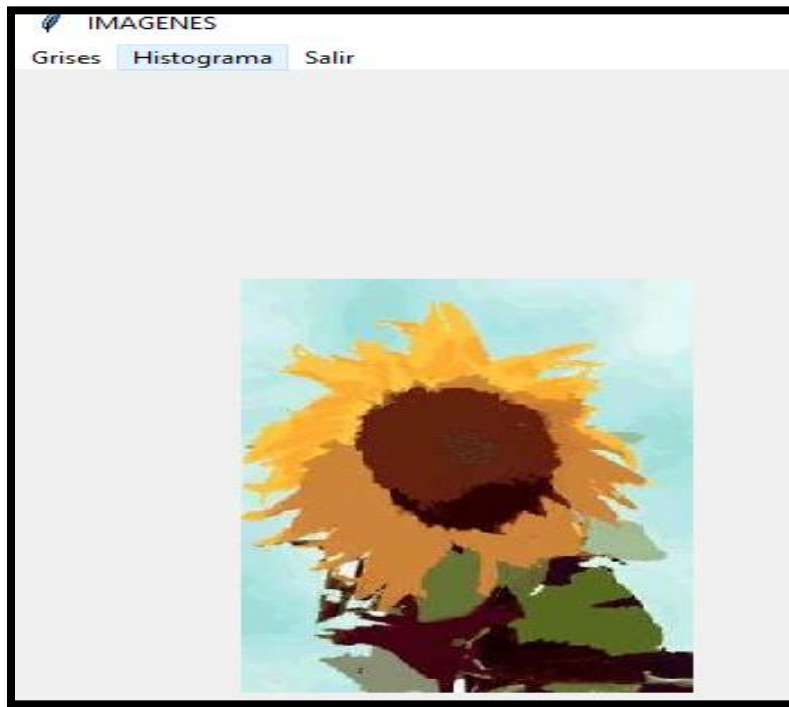
Figura3:Imágenes de PDI

Diseñar y programar un script que permita realizar las siguientes tareas

- Leer las imágenes "sunflower.zip"
  - ❖ Directorio de donde vamos a leer las imágenes



- ❖ Las imágenes las puedes escoger y serán visualizadas para después aplicar el procedimiento

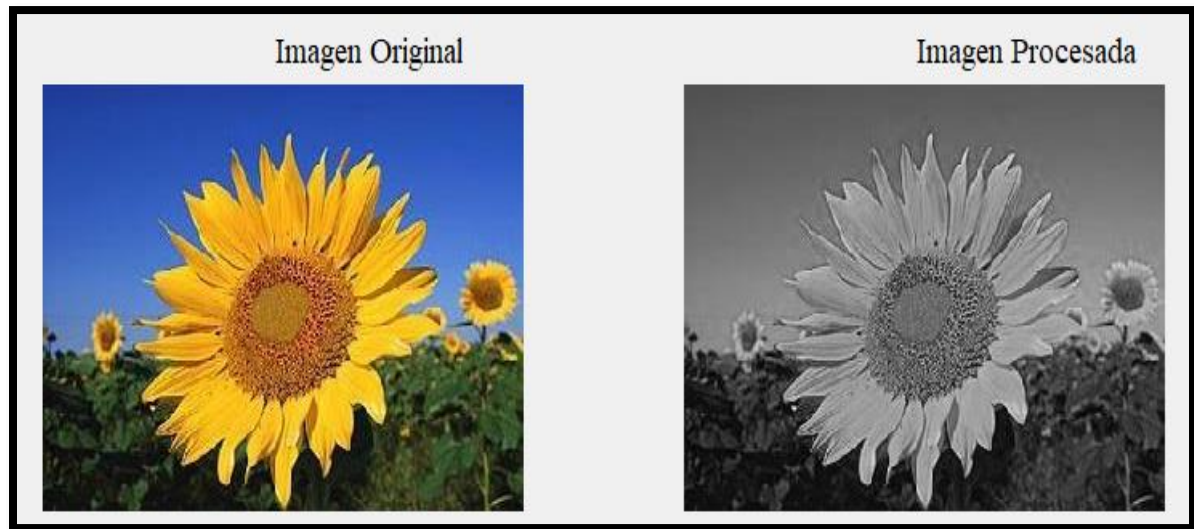


- ❖ Nivel de código desarrollado en Python

```
def abrir():
    varios=85#declaracion de variables
    promedio=0#declaracion de variables
    tiempo_inicial = time() #inicia el tiempo para calcular
    ventana.filename = filedialog.askopenfilename(initialdir = 'C:/Users/chris/OneDrive/Desktop/Procemiento/imagenes',
    title = "Elige Tu Archivo De Imagen:",
    filetypes = (("Imagenes PNG", "*.png"),("Imagenes GIF ", "*.gif"))) # creamos un ventana y podemos escogerla ruta
    global ruta#inicialimos la variable global
    ruta = ventana.filename
    imagenL = PhotoImage(file = ruta)#la ruta le pasamos a un imagen
    global abrirImagen#inicialimos la variable global
    abrirImagen = canvas.create_image(100, 150, anchor=NW, image=imagenL)#Abrimos las ventanas
    ventana.mainloop()#visualizamos la ventan
    tiempo_final = time()#calculamos el tiempo final
    tiempo_ejecucion = tiempo_final - tiempo_inicial#calculamos el tiempo
    print('*****')
    print('*****')
    print('El tiempo de ejecucion fue imagen a color:',tiempo_ejecucion ) #En segundos
    promedio=tiempo_ejecucion*varios#sacamos el promedio de todas la imagenes
    print('El tiempo de ejecucion fue promedio de images totales:',promedio )
```

- **Conversión a escala de grises**

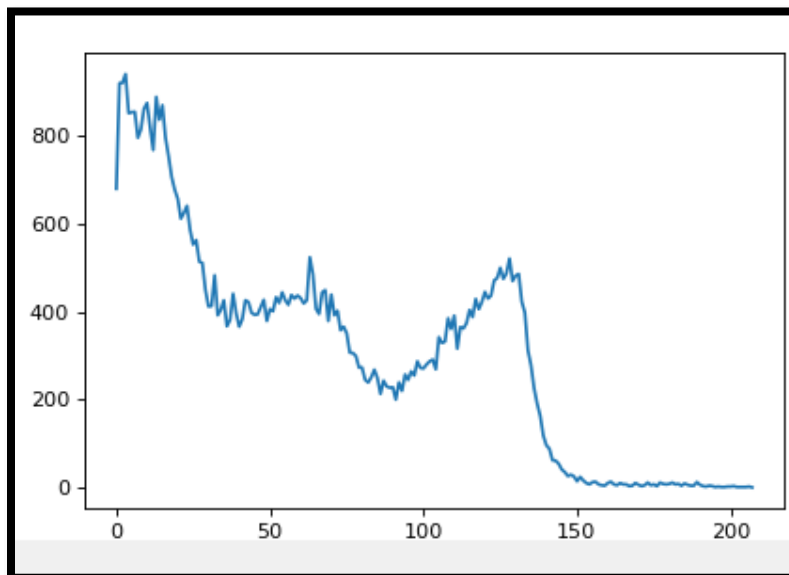
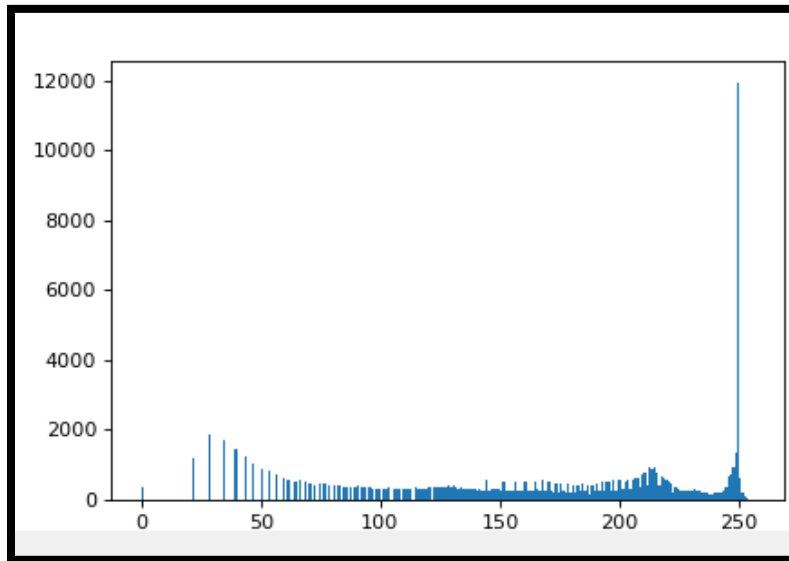
- ❖ Después que escogemos la imagen realizamos el proceso de convertir en escala de grises
- ❖ En la primera imagen podemos ver la imagen Original
- ❖ En la segunda la imagen en grises



- ❖ Visualizaremos a nivel de código el proceso gris

```
def grises():
    varios1=85# declamos la variables
    promediol=0# declamos la variables
    tiempo_inicial1 = time() #inicia el tiempo para calcular
    im = Image.open(ruta)#la imagen de fue abierta la original y se guarda de formayo gif la recuperamos para poder realizar el proceso
    im2 = im
    i = 0
    while i < im2.size[0]:#aplicamos una condicion
        j = 0#declaracion de variables
        while j < im2.size[1]:#aplicamos una condicion
            r, g, b = im2.getpixel((i, j))#realizamos r1 proceso de convertir
            g = (r + g + b) / 3#realizamos r1 proceso de convertir
            gris = int(g)#realizamos r1 proceso de convertir
            pixel = tuple([gris, gris, gris])#realizamos r1 proceso de convertir
            im2.putpixel((i, j), pixel)#realizamos r1 proceso de convertir
            j+=1
        i+=1
    g = im2.convert('L')
    g.save('C:/Users/chris/OneDrive/Desktop/Procemiento/resultados/grises.gif') #gurdarnamso la imagen que fur procesda
    imagenL = PhotoImage(file = 'C:/Users/chris/OneDrive/Desktop/Procemiento/resultados/grises.gif')#gurdarnamso la imagen que fur procesda
    global grisesito#declaracion de variable
    grisesito = canvas.create_image(500, 150, anchor=NW, image=imagenL)
    ventana.mainloop()#abrimos la ventana
    tiempo_final1 = time() #caculamos el tiempo final
    tiempo_ejecucion1 = tiempo_final1 - tiempo_inicial1#calculamos el tiempo
    print('El tiempo de ejecucion fue imagen a gris:',tiempo_ejecucion1 ) #calculamos el tiempo
    promediol=tiempo_ejecucion1*varios1#sacamos el promedio de todas la imagenes
    print('El tiempo de ejecucion fue promedio de imagenes totales griss:',promediol )#calculamos el promedio
```

- **Cálculo de los histogramas a color y en escala de grises**
  - ❖ Histograma original
  - ❖ Histograma procesamiento gris





```

def histograma():
    im = Image.open("C:/Users/chris/OneDrive/Desktop/Procemiento/resultados/grises.gif")#Abrimos la imagen
    im16 = im#declaramos las variblas
    [ren, col] = im16.size#sacamos el tamaño
    total = ren * col#realizamo la multiplicacion

    a = np.asarray(im16, dtype = np.float32)#agregamos un array
    a = a.reshape(1, total)
    a = a.astype(int)
    a = max(a)#sacamos los valores maximo
    valor = 0
    maxd = max(a)
    grises = maxd
    vec=np.zeros(grises + 1)#Devuelve una nueva matriz de forma y tipo dados, llena de ceros.
    for i in range(total - 1):#declaramos el for y inicializamos el rango
        valor = a[i]
        vec[valor] = vec[valor] + 1
    plt.plot(vec)#graficamos
    plt.savefig('C:/Users/chris/OneDrive/Desktop/Procemiento/resultados/histograma.png', dpi=80)
    imagenL = PhotoImage(file = 'C:/Users/chris/OneDrive/Desktop/Procemiento/resultados/histograma.png')
    global hist#declaracion de variable
    hist = canvas.create_image(280, 355, anchor=NW, image=imagenL)
    ventana.mainloop()

#####
#####
def histogramaori():
    img = cv2.imread('C:/Users/chris/OneDrive/Desktop/Procemiento/imagenes/grises.png',0)#abrimos la imagen de que fe guardar con aterioridad
    plt.hist(img.ravel(),256,[0,256])#realizamos la grafica
    plt.savefig('C:/Users/chris/OneDrive/Desktop/Procemiento/propios/histogramas.png', dpi=80)#guardamos la inagemn
    imagenL1 = PhotoImage(file = 'C:/Users/chris/OneDrive/Desktop/Procemiento/propios/histogramas.png')
    global hist1#declaracion de variables
    hist1 = canvas.create_image(280, 355, anchor=NW, image=imagenL1)
    ventana.mainloop()#visualizamo sl imgane

#####
#####
def limniar():

```

- Tiempo que toma procesar todas las imágenes

```

El tiempo de ejecución promedio una imagen a griss : 7.310272216796875
El tiempo de ejecución las imágenes totales griss: 621.3731384277344
*****
*****
El tiempo de ejecución promedio una imagen a color : 12.44281530380249
El tiempo de ejecución las imágenes totales color: 1057.6393008232117

```

- Tiempo promedio de procesamiento de imágenes

```

El tiempo de ejecución promedio una imagen a griss : 7.310272216796875
El tiempo de ejecución las imágenes totales griss: 621.3731384277344
*****
*****
El tiempo de ejecución promedio una imagen a color : 12.44281530380249
El tiempo de ejecución las imágenes totales color: 1057.6393008232117

```

- **Tamaño inicial de las imágenes (color) frente al tamaño final (escala de grises)**
  - ❖ Su tamaño tiene un pequeño variante de la original con respecto a la imagen procesada como puede visualizar



## Linux Mint

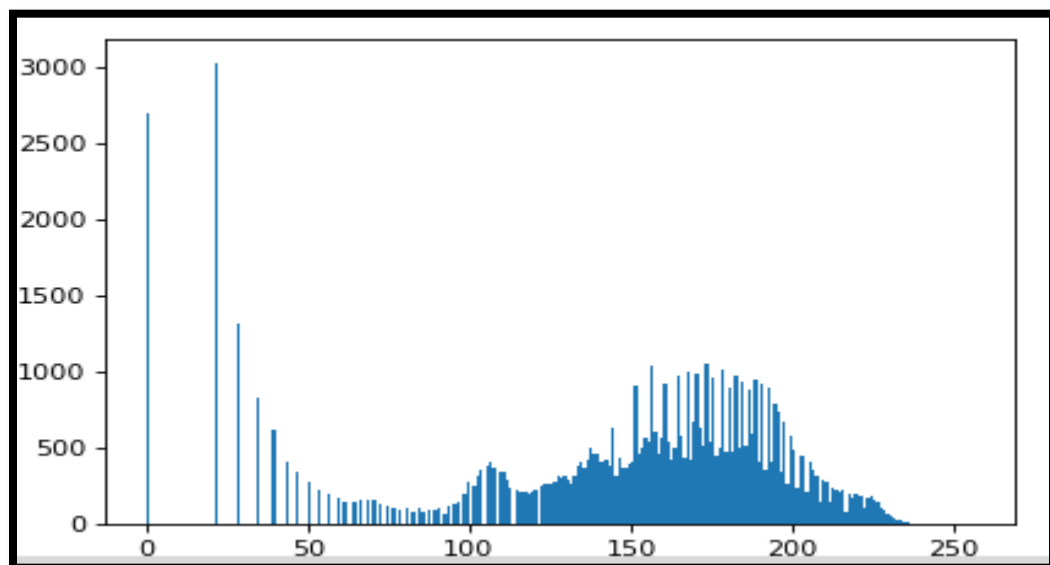
- ❖ Descripción del sistema

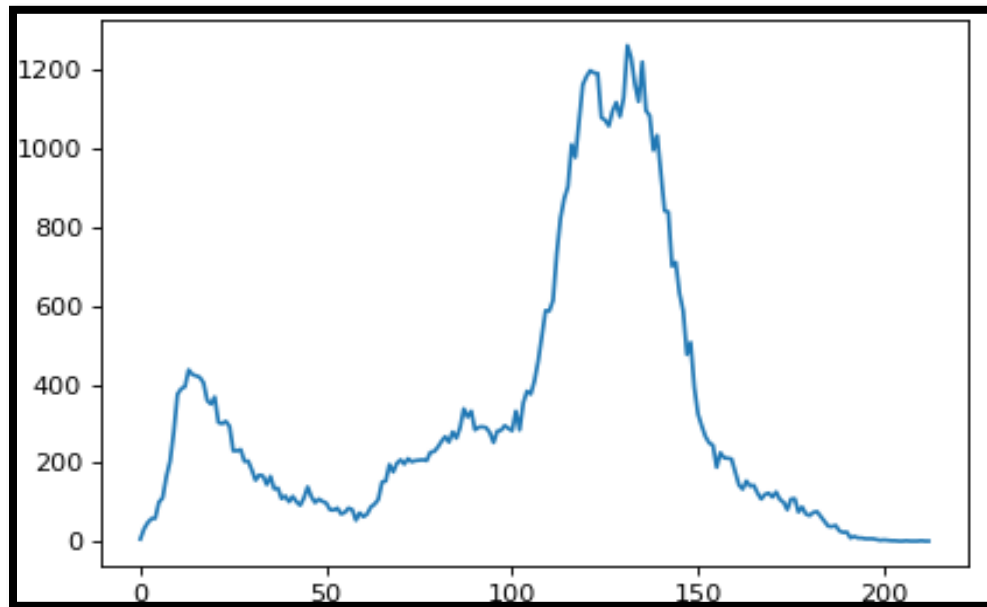
```
cris@cris-VirtualBox:~$ cat /etc/linuxmint//info
RELEASE=20
CODENAME=ulyana
EDITION="Cinnamon"
DESCRIPTION="Linux Mint 20 Ulyana"
DESKTOP=Gnome
TOOLKIT=GTK
NEW_FEATURES_URL=https://www.linuxmint.com/rel_ulyana_cinnamon_whatsnew.php
RELEASE_NOTES_URL=https://www.linuxmint.com/rel_ulyana_cinnamon.php
USER_GUIDE_URL=https://www.linuxmint.com/documentation.php
GRUB_TITLE=Linux Mint 20 Cinnamon
```

- ❖ Realizamos el mismo proceso donde cargamos la imagen original y también el proceso de convertir a escala de grises, pero en Linux Mint



- **Cálculo de los histogramas a color y en escala de grises Linux Mint**
  - ❖ Histograma original
  - ❖ Histograma procesamiento gris





- Tiempo que toma procesar todas las imágenes Linux Mint

```
El tiempo de ejecucion promedio una imagen a griss : 3.344888687133789
El tiempo de ejecucion las images totales griss: 284.31553840637207
*****
*****
El tiempo de ejecucion promedio una imagen a color : 9.09675908088684
El tiempo de ejecucion las images totales color: 773.2245218753815
```

- Tiempo promedio de procesamiento de imágenes Linux Mint

```
El tiempo de ejecucion promedio una imagen a griss : 3.344888687133789
El tiempo de ejecucion las images totales griss: 284.31553840637207
*****
*****
El tiempo de ejecucion promedio una imagen a color : 9.09675908088684
El tiempo de ejecucion las images totales color: 773.2245218753815
```

- ❖ **Como podemos observar Linux es mas eficiente a la hora de realizar la conversión con un directorio de imágenes de forma grises**

### **Referencia**

- ❖ Cuesta, U., Niño, J. y Rodríguez, J. (2017). The Cognitive Processing of an Educational App with Electroencephalogram and “Eye Tracking”. *Comunicar*, 25(52), 41-50. DOI: 10.3916/ C52-2017-04
- ❖ Dalal, N., y Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. En 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) (Vol. 1, 886-893). IEEE. DOI: 10.1109/CVPR.2005.177
- ❖ Euromonitor International. (Agosto del 2017). Digital Consumer Profiles: How Latin Americans will shop and spend digitally. Análisis del consumidor. Base de datos Euromonitor International.
- ❖ Hidalgo, I., y Sánchez, R. (2015). Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales (Trabajo de fin de grado). Universidad Complutense de Madrid, España.
- ❖ Kaur, S. (2016). An automatic number plate recognition system under image processing. *International Journal of Intelligent Systems and Applications*, 8(3), 14-25. DOI: 10.5815/ijisa.2016.03.02