

# UNIVERSIDAD POLITECNICA SALESIANA

Nombre: Marcela Zhagüi

TIEMPO TOTAL DE PROCESAMIENTO: 63.07405877113342 s.

TIMEPO PROMEDIO GRISES: 3.192654848098755 s.

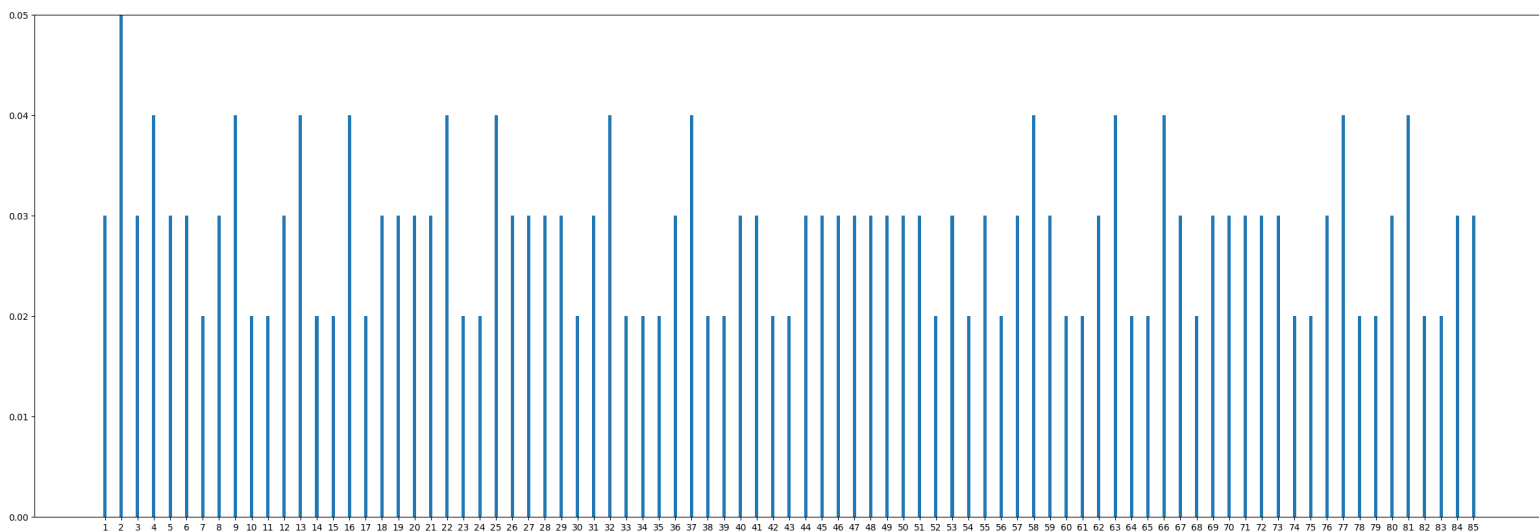
TIMEPO PROMEDIO del histograma de imágenes originales: 57.50421857833862 s.

TIMEPO PROMEDIO para el histograma en escala de grises: 2.3719890117645264 s.

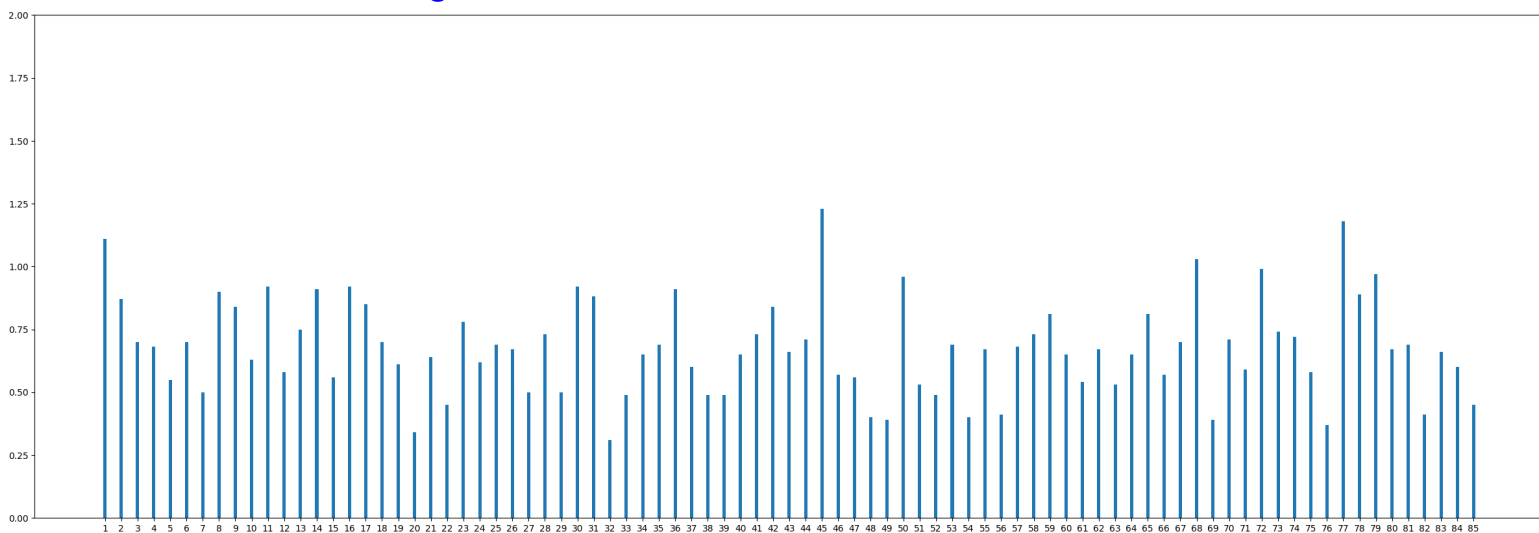
PESO ORIGINAL:1.2289628982543945

PESO DE ESCALA DE GRISES: 0.9997129440307617

## HISTOGRAMAS



Histograma de 1 canal



Histograma de 3 canales

# Script

```
import os
import time
from reportlab.lib.utils import ImageReader
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import A4
import pylab as pl
import numpy as np
from time import sleep

conversion_to_gris_time = 0
histograma3canales_time = 0
histograma1canal_time = 0

inicio = time.time()
imagenes= os.listdir('sunflower/')
sunflowers_gris = os.listdir('sunflower-gris/')

diccionario_colores = {}
diccionario_grises = {}

def convertir_to_gray(imagenes, conversion_to_gris_time):
    ## leer archivo
    print('_____')
    print("Convirtiendo imágenes a escala de grises")
    print('-----')

    n = 0;
    for j in imagenes:
        inicio_gris = time.time()
        orden = 'convert \'sunflower/' + j \
            + '\' -set colorspace Gray -separate -average \'sunflower-gris/' \
            + j + '\"
        os.system(orden)
        conversion_to_gris_time += (time.time()-inicio_gris)
        n += 1
    print('Tiempo promedio para convertir a escala de grises',n, '::::::\')
```

```

    ,str(conversion_to_gris_time/n), 'segundos.')
print("Conversión finalizada")
return conversion_to_gris_time

```

```

def histograma_1_canal(imagenes, histograma1canal_time, diccionario_grises):
    print('-----')
    print("CONvirtiendO histogramas de las imágenes en escala de grises")
    cont = 1
    for imagen in imagenes:
        start = time.time()
        carpeta = imagen[0: len(imagen)-4: 1]
        orden = 'convert sunflower/' + imagen + ' -colorspace Gray -define histogram:unique-'
        os.system(orden)
        os.system('mv histograma-' + carpeta + '.gif histogramas-canal1/')
        diccionario_grises[str(cont)] = round((time.time()-start),2)
        histograma1canal_time += (time.time()-start)
        cont += 1
    print('Tiempo Promedio para obtener histogramas en gris ----->', cont, ':' '\n'
        ,str(histograma1canal_time/cont), ' segundos.')
    print('-----')
    print("Resultado de Histogramas")
    print('-----')
    return histograma1canal_time

```

```

def histograma_3_canales(imagenes, histograma3canales_time, diccionario_colores):
    print('-----')
    print("Obteniendo los histogramas de las imágenes a color...")
    print('-----')
    cont = 1
    for k in imagenes:
        start_color = time.time()
        carpeta = k[0: len(k)-4: 1]
        existe = os.path.exists('histogramas-canal3/Histogramas-' + carpeta)
        if existe != True:
            os.system('mkdir histogramas-canal3/Histogramas-' + carpeta)
        orden = 'convert sunflower/' + k + ' -define histogram:unique-colors=true -format %c h

```

```

    + carpeta + '/histograma.gif'
os.system(orden)
orden = 'convert histogramas-canal3/Histogramas-'+carpeta+'/histograma.gif -strip
    + carpeta+'/canal-%d.gif'
os.system(orden)
diccionario_colores[str(cont)] = round((time.time()-start_color),2)
histograma3canales_time += (time.time()-start_color)
cont += 1
print('Tiempo promedio para obtener histogramas a color',cont ,'::::\n
    ,str(histograma3canales_time/cont), 'segundos.')
print('-----')
print("Histogramas obtenidos")
print('-----')
return histograma3canales_time

```

```

def size(ruta):
    img = 0
    for path, dirs, files in os.walk(ruta):
        for archivo in files:
            img += os.path.getsize(os.path.join(path, archivo))
    return img

```

```

def graficar_grises(diccionario_grises):
    f, ax = pl.subplots(figsize=(30,10))
    x = np.arange(len(diccionario_grises))
    pl.bar(x, diccionario_grises.values(), align='center', width=0.2)
    pl.xticks(x, diccionario_grises.keys())
    ymax = 0.05
    pl.ylim(0, ymax)
    pl.savefig('Figura_1', bbox_inches='tight',pad_inches=0.1)

```

```

def graficar_colores(diccionario_colores):
    f, ax = pl.subplots(figsize=(30,10))
    x = np.arange(len(diccionario_colores))
    pl.bar(x, diccionario_colores.values(), align='center', width=0.2)
    pl.xticks(x, diccionario_colores.keys())

```

```
ymin = 0
ymax = 2
pl.ylim(0, ymax)
pl.savefig('Figura_2', bbox_inches='tight', pad_inches=0.1)
```

```
conversion_to_gris_time = convertir_to_gray(imagenes, conversion_to_gris_time)
```

```
histograma3canales_time = histograma_3_canales(imagenes, histograma3canales_time)
histograma1canal_time = histograma_1_canal(sunflowers_gray, histograma1canal_time)
```

```
peso_original = size('sunflower/')/1048576
print('PESO IMAGEN ORIGINAL: ' , peso_original, 'MB')
```

```
peso_gris = size('sunflower-gris/')/1048576
print('PESO IMAGENES EN GRIS: ' , peso_gris, 'MB')
```

```
tiempo_total = (time.time()-inicio)
print('TIEMPO total de ejecucion', tiempo_total, 'segundos.')
```

```
graficar_grises(diccionario_grises)
graficar_colores(diccionario_colores)
```

```
##_____INFORME_____
```

```
print('Generando PDF')
documento = canvas.Canvas('Practica0_ZHAGUI_MARCELA', pagesize=A4)
documento.setFont("Helvetica", 23)
documento.setFillColor('blue')
documento.drawString(100,800,'UNIVERSIDAD POLITECNICA SALESIANA')
documento.borderColor = 'black'
documento.setFont("Helvetica", 15)
documento.setFillColor('black')
x = 15
y = 770
```

```
documento.drawString(x,y,'Nombre: Marcela Zhagüi')
documento.drawString(x,y-20,'TIEMPO TOTAL DE PROCESAMIENTO: '+str(tiempo_to
documento.drawString(x,y-40,'TIMEPO PROMEDIO GRISES: '+str(conversion_to_gris_
```

```

documento.drawString(x,y-60,'TIMEPO PROMEDIO del histograma de imágenes origin
documento.drawString(x,y-80,'TIMEPO PROMEDIO para el histograma en escala de gr
documento.drawString(x,y-100,'PESO ORIGINAL:' +str(peso_original))
documento.drawString(x,y-130,'PESO DE ESCALA DE GRISES: ' +str(peso_gris))
documento.setFont("Helvetica", 23)
documento.setFillColor('black')
documento.drawString(x+240,y-180,'HISTOGRAMAS')
documento.drawImage("Figura_1.png", 10, y-400, width=580, height=200)
documento.setFont("Helvetica", 15)
documento.setFillColor('blue')
documento.drawString(x+120,y-425,'Histograma de 1 canal')
documento.drawImage("Figura_2.png", 10, y-630, width=580, height=200)
documento.drawString(x+120,y-650,'Histograma de 3 canales')
documento.setFillColor('black')

documento.showPage()
documento.setFont("Helvetica", 23)
documento.setFillColor('blue')
documento.drawString(210,800,'Script')
documento.setFont("Helvetica", 15)
documento.setFillColor('black')
doc = open("practica0_imagenes.py","r")

lineas = doc.readlines()
doc.close()
for linea in lineas:
    y = y-20
    if y < 40:
        documento.showPage()
        documento.setFont("Helvetica", 15)

        y = 770
        documento.drawString(x,y,linea[0:len(linea)-1])

documento.save()
print('proceso terminado...')

```