

Trabalho 1 - Tipo de Imagem: TC

Descrição do trabalho:

<http://webserver2.tecgraf.puc-rio.br/~mgattass/visao/trb/T1.html> (<http://webserver2.tecgraf.puc-rio.br/~mgattass/visao/trb/T1.html>).

Aluno: Daniel da Silva Costa

E-mail: danieldasilvacosta@gmail.com

Mudando o diretório para o meu Google Drive.

In [1]:

```
from google.colab import drive
drive.mount('/content/drive/')
```

Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive.mount("/content/drive/", force_remount=True).

In [2]:

```
cd "drive/MyDrive/Doutorado/Disciplinas/[2022.2] [PUC-Rio] Visão Computacional - Professor Marcelo Gattass/Trabalhos/Trabalho 1/"
```

```
/content/drive/MyDrive/Doutorado/Disciplinas/[2022.2] [PUC-Rio] Visão Computacional - Professor Marcelo Gattass/Trabalhos/Trabalho 1
```

In [3]:

```
!pwd
```

```
/content/drive/MyDrive/Doutorado/Disciplinas/[2022.2] [PUC-Rio] Visão Computacional - Professor Marcelo Gattass/Trabalhos/Trabalho 1
```

Imports

In [4]:

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

from skimage import io
```

In [5]:

```
path = './imagens/tc/'
```

In [6]:

```
img = io.imread( path + 'antes0900.tiff' )  
print( img.shape )
```

(925, 920)

In [7]:

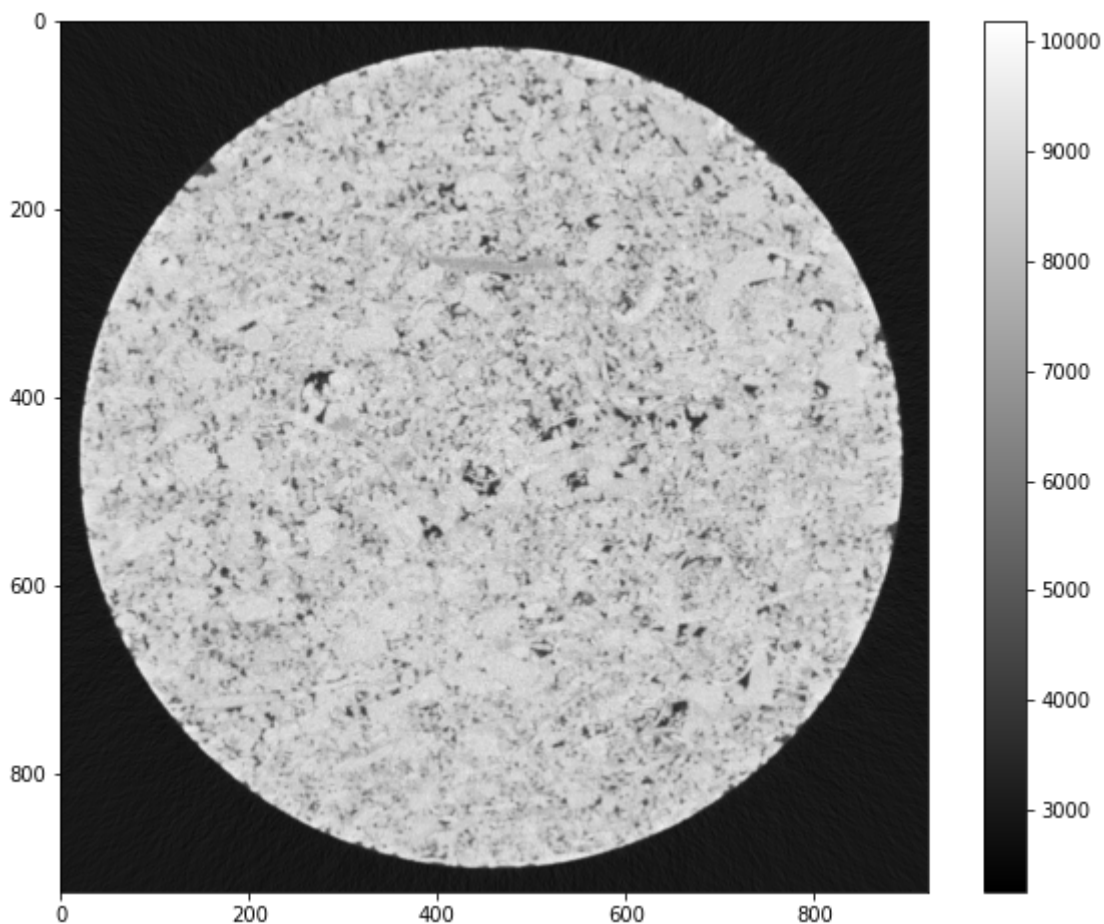
```
img
```

Out[7]:

```
array([[2755, 2831, 3188, ..., 3022, 3115, 3140],  
       [3069, 2848, 3091, ..., 2988, 3125, 3154],  
       [3066, 2981, 3074, ..., 3057, 3239, 2980],  
       ...,  
       [2986, 3102, 3109, ..., 2927, 2980, 3155],  
       [3056, 3133, 2796, ..., 3133, 2916, 3065],  
       [3053, 2865, 2813, ..., 2927, 2990, 2958]], dtype=uint16)
```

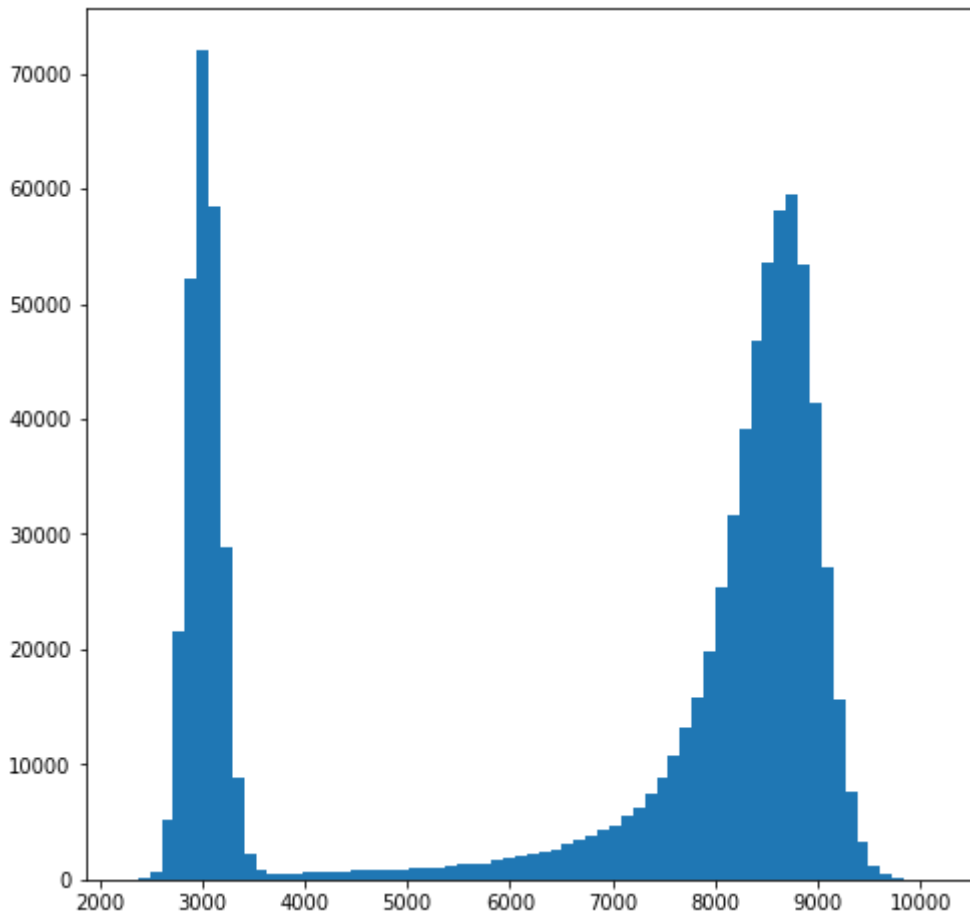
In [8]:

```
plt.figure( figsize=( 10, 8 ) )  
plt.imshow( img, cmap='gray' )  
plt.colorbar()  
plt.show()
```



In [9]:

```
plt.figure( figsize=(8, 8) )  
n, bins, patches = plt.hist( img.ravel(), bins='auto' )  
plt.show()
```



In [10]:

```
import statistics  
  
values = img.ravel()  
values = values[ values >= 3500 ]  
  
print( f'Média: {statistics.mean( values )}')  
print( f'Mediana: {statistics.median( values )}')  
print( f'Moda: {statistics.mode( values )}')
```

Média: 8278

Mediana: 8508

Moda: 8725

In [11]:

```
print( n.shape, bins.shape )
print( f'n = {n}' )
print( f'bins = {bins}' )
```

```
(69,) (70,)
n = [7.0000e+00 6.2000e+01 7.4400e+02 5.1900e+03 2.1518e+04 5.2181e+04
7.2100e+04 5.8457e+04 2.8776e+04 8.8120e+03 2.1770e+03 7.5200e+02
4.7800e+02 4.5300e+02 5.3800e+02 6.0600e+02 6.0800e+02 6.6600e+02
6.9700e+02 7.6600e+02 7.9200e+02 8.0800e+02 9.1300e+02 8.7600e+02
9.5200e+02 1.0190e+03 1.0600e+03 1.2470e+03 1.3100e+03 1.4270e+03
1.4350e+03 1.6810e+03 1.8430e+03 1.9940e+03 2.1490e+03 2.4460e+03
2.6050e+03 3.0200e+03 3.4140e+03 3.7900e+03 4.2470e+03 4.6770e+03
5.5780e+03 6.1750e+03 7.4270e+03 8.8910e+03 1.0702e+04 1.3128e+04
1.5827e+04 1.9832e+04 2.5351e+04 3.1606e+04 3.9100e+04 4.6747e+04
5.3505e+04 5.8070e+04 5.9478e+04 5.3324e+04 4.1329e+04 2.7208e+04
1.5577e+04 7.6140e+03 3.2880e+03 1.2420e+03 4.7400e+02 1.6600e+02
4.8000e+01 1.6000e+01 4.0000e+00]
bins = [ 2259.          2373.85507246  2488.71014493  2603.56521739
2718.42028986  2833.27536232  2948.13043478  3062.98550725
3177.84057971  3292.69565217  3407.55072464  3522.4057971
3637.26086957  3752.11594203  3866.97101449  3981.82608696
4096.68115942  4211.53623188  4326.39130435  4441.24637681
4556.10144928  4670.95652174  4785.8115942   4900.66666667
5015.52173913  5130.37681159  5245.23188406  5360.08695652
5474.94202899  5589.79710145  5704.65217391  5819.50724638
5934.36231884  6049.2173913   6164.07246377  6278.92753623
6393.7826087   6508.63768116  6623.49275362  6738.34782609
6853.20289855  6968.05797101  7082.91304348  7197.76811594
7312.62318841  7427.47826087  7542.33333333  7657.1884058
7772.04347826  7886.89855072  8001.75362319  8116.60869565
8231.46376812  8346.31884058  8461.17391304  8576.02898551
8690.88405797  8805.73913043  8920.5942029   9035.44927536
9150.30434783  9265.15942029  9380.01449275  9494.86956522
9609.72463768  9724.57971014  9839.43478261  9954.28985507
10069.14492754 10184.          ]
```

In [12]:

```
prefix = 'antes0'
vol = np.zeros( (13, 925, 920), dtype='uint16' ) # número de elementos x altura x largu
ra

for i in range(900, 913):

    fname = path + prefix + f'{i}.tiff'
    vol[ i-900, :, : ] = io.imread( fname )

vol
```

Out[12]:

```

array([[2755, 2831, 3188, ..., 3022, 3115, 3140],
       [3069, 2848, 3091, ..., 2988, 3125, 3154],
       [3066, 2981, 3074, ..., 3057, 3239, 2980],
       ...,
       [2986, 3102, 3109, ..., 2927, 2980, 3155],
       [3056, 3133, 2796, ..., 3133, 2916, 3065],
       [3053, 2865, 2813, ..., 2927, 2990, 2958]],

       [[2909, 2785, 3028, ..., 2855, 3057, 3295],
       [3091, 2835, 2852, ..., 2843, 3309, 2996],
       [3028, 3027, 2810, ..., 3076, 3067, 2707],
       ...,
       [3227, 3154, 3131, ..., 3095, 3054, 3051],
       [2977, 3008, 2913, ..., 3128, 3098, 3141],
       [2926, 2888, 2902, ..., 3044, 3011, 3067]],

       [[2932, 2965, 2891, ..., 2766, 2950, 3218],
       [2927, 2914, 2799, ..., 2979, 3172, 3096],
       [3091, 3037, 2902, ..., 2983, 3054, 2978],
       ...,
       [3257, 3076, 3074, ..., 3046, 3041, 2848],
       [3157, 2941, 2842, ..., 3193, 3032, 2999],
       [2833, 2736, 2787, ..., 3145, 3244, 3078]],

       ...,

       [[3053, 3070, 3206, ..., 2736, 2873, 2994],
       [2803, 2873, 2876, ..., 2895, 2983, 3017],
       [3058, 2986, 2878, ..., 2977, 3011, 3026],
       ...,
       [3063, 2935, 2969, ..., 2673, 2890, 3071],
       [2929, 2957, 2821, ..., 3169, 2874, 3074],
       [3005, 2902, 3040, ..., 3443, 3137, 3087]],

       [[3048, 3077, 3084, ..., 3029, 2887, 2964],
       [2934, 3043, 2975, ..., 2890, 2984, 3040],
       [2962, 3012, 3055, ..., 3021, 2992, 2992],
       ...,
       [3134, 3221, 3124, ..., 3072, 3020, 2944],
       [3163, 2989, 2984, ..., 3115, 3154, 3241],
       [3044, 2988, 3103, ..., 3280, 3109, 3168]],

       [[2962, 2968, 3055, ..., 3305, 2938, 2951],
       [2923, 2987, 2980, ..., 2913, 2989, 2890],
       [3084, 3059, 3060, ..., 3130, 2941, 2965],
       ...,
       [3218, 3204, 3172, ..., 3106, 2954, 2873],
       [3286, 3157, 3031, ..., 2959, 3136, 3072],
       [3184, 2964, 3137, ..., 3178, 2983, 2999]]], dtype=uint16)

```

In [13]:

```
vol[0]
```

Out[13]:

```
array([[2755, 2831, 3188, ..., 3022, 3115, 3140],
       [3069, 2848, 3091, ..., 2988, 3125, 3154],
       [3066, 2981, 3074, ..., 3057, 3239, 2980],
       ...,
       [2986, 3102, 3109, ..., 2927, 2980, 3155],
       [3056, 3133, 2796, ..., 3133, 2916, 3065],
       [3053, 2865, 2813, ..., 2927, 2990, 2958]], dtype=uint16)
```

In [14]:

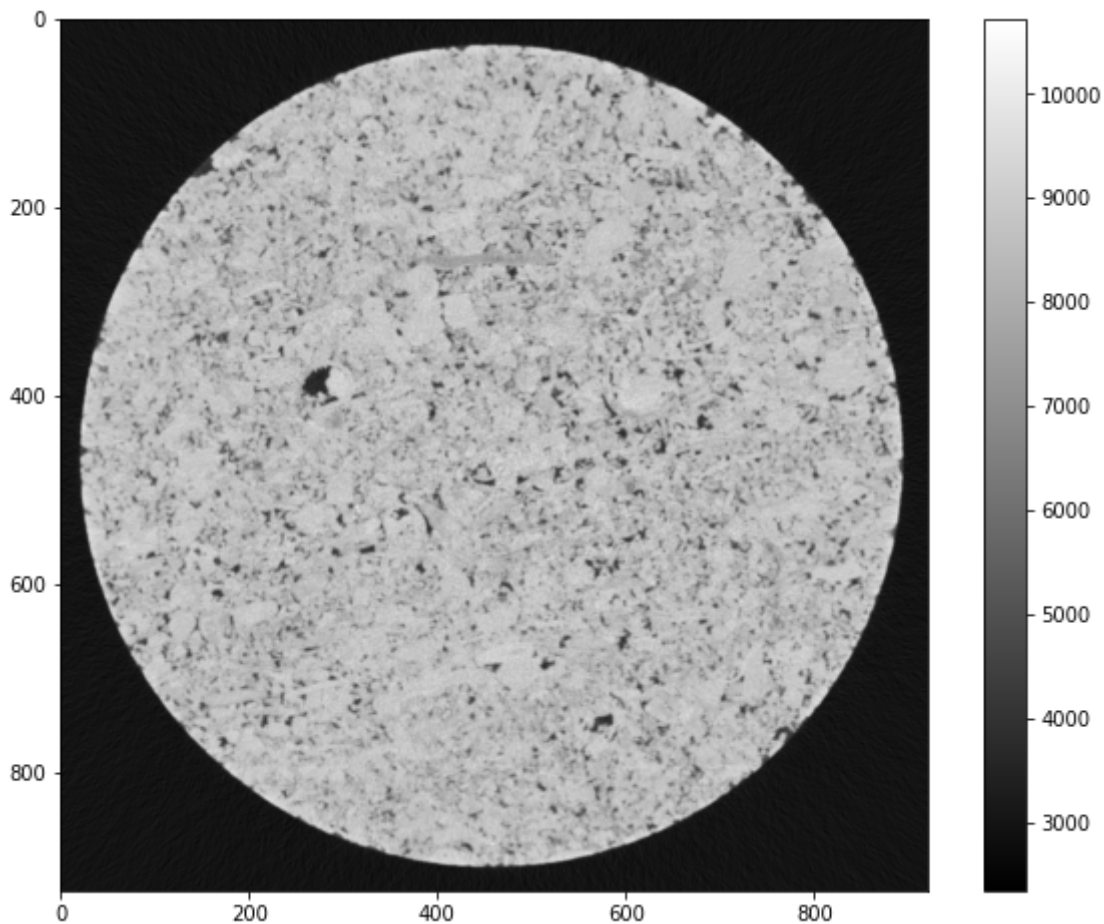
```
len(vol)
```

Out[14]:

13

In [15]:

```
plt.figure( figsize=( 10, 8 ) )
plt.imshow( vol[12, :], cmap='gray' )
plt.colorbar()
plt.show()
```



In [16]:

```
def scale( data, x_min, x_max, newx_min, newx_max ):

    data = data.astype( np.float32 )
    data = (data - x_min) * ( (newx_max - newx_min) / (x_max - x_min) ) + newx_min
    data[ data > newx_max ] = newx_max
    data[ data < newx_min ] = newx_min

    return data
```


In [17]:

```
vol_f = scale( vol, 4000, 10000, 0, 1 )
vol_f
```

Out[17]:

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]],

       ...,

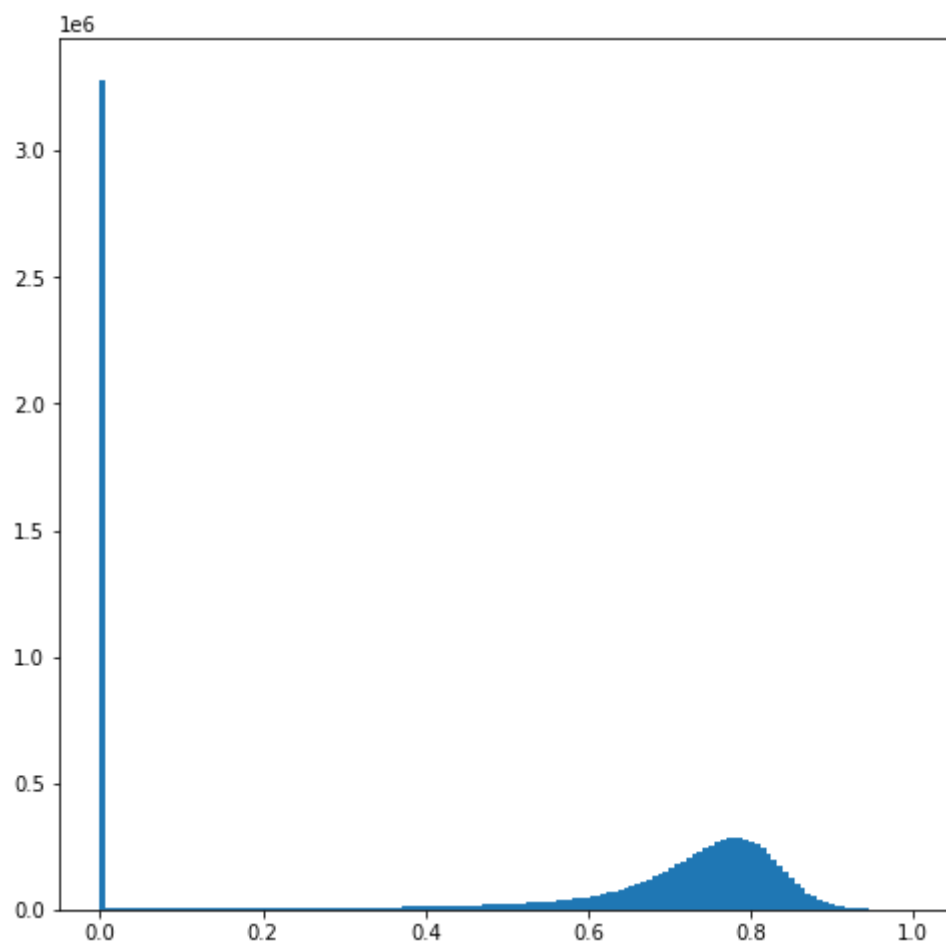
       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]],

       [[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]], dtype=float32)
```

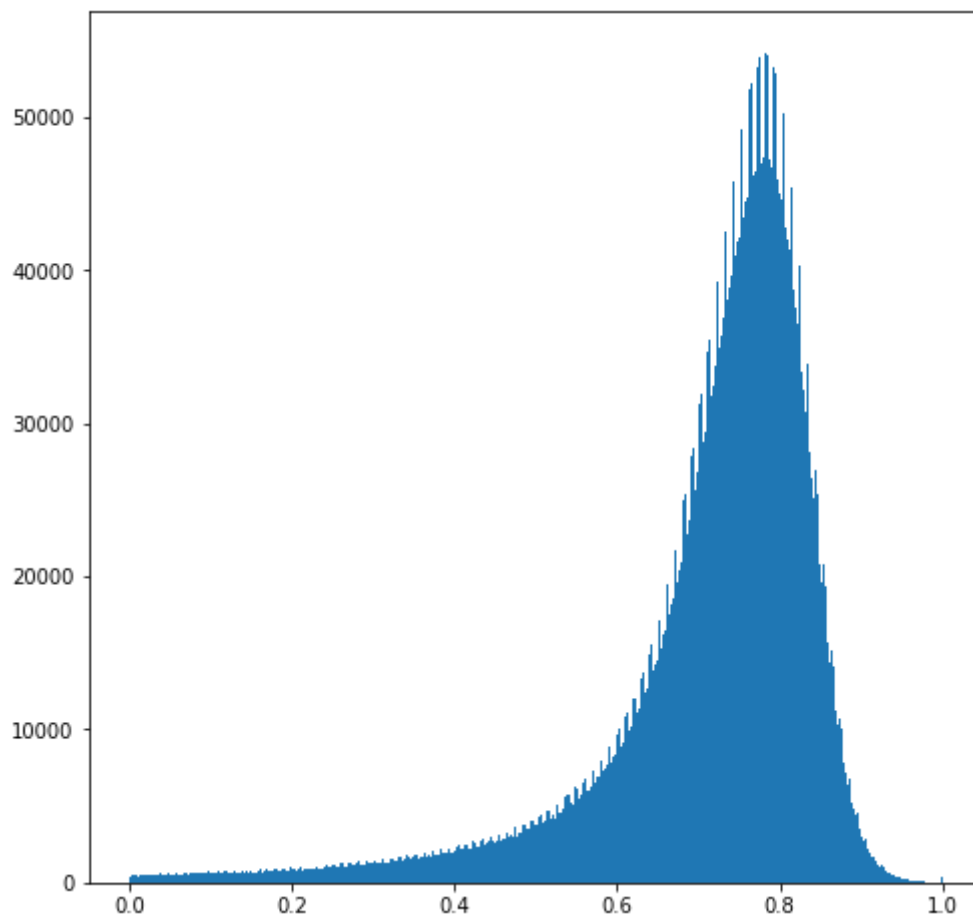
In [18]:

```
data = vol_f  
plt.figure( figsize=(8, 8) )  
n, bins, patches = plt.hist( data.ravel(), bins='auto' )  
plt.show()
```



In [19]:

```
data = vol_f  
data = data[ data > 0 ]  
plt.figure( figsize=(8, 8) )  
n, bins, patches = plt.hist( data.ravel(), bins='auto' )  
plt.show()
```



In [20]:

```
data = vol_f  
data = data[ data > 0 ]
```

In []:

```
%%time

import statistics

values = data.ravel()
values = values[ values > 0 ]

mean_value = statistics.mean( values )
median_value = statistics.median( values )
mode_value = statistics.mode( values )

print( f'Média: {mean_value}')
print( f'Mediana: {median_value}')
print( f'Moda: {mode_value}')
```

Média: 0.7160080075263977
Mediana: 0.7508333325386047
Moda: 0.7761666774749756
CPU times: user 18.8 s, sys: 218 ms, total: 19 s
Wall time: 19 s

In [22]:

```
def get_y( x ):

    n, bins, patches = plt.hist( values, bins='auto' )

    greater_bin = -1
    for index, bin in enumerate( bins ):

        if bin <= x:

            greater_bin = bins[ index + 1 ]
            index_bin = index + 1

    print( f'greater_bin: {greater_bin}' )
    print( f'index_bin: {index_bin}' )

    y = n[ index_bin ]
    print( f'y: {y}' )
    print()

    return y
```

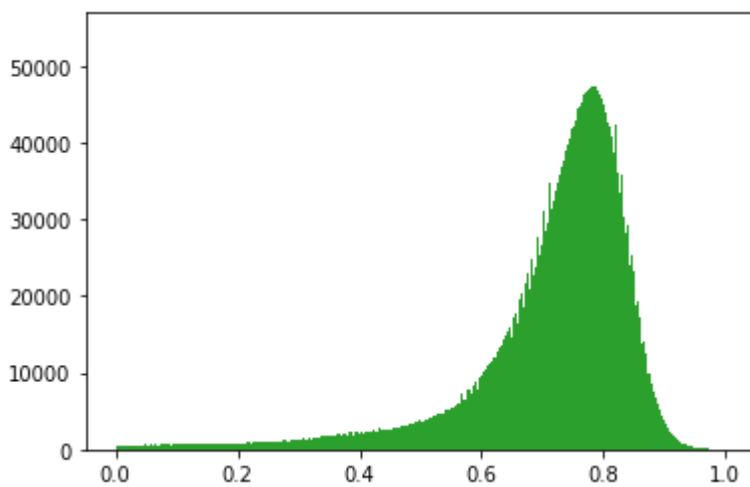
In [23]:

```
y_mean = get_y( mean_value )  
y_median = get_y( median_value )  
y_mode = get_y( mode_value )
```

```
greater_bin: 0.7171443104743958  
index_bin: 583  
y: 36880.0
```

```
greater_bin: 0.7515789270401001  
index_bin: 611  
y: 42731.0
```

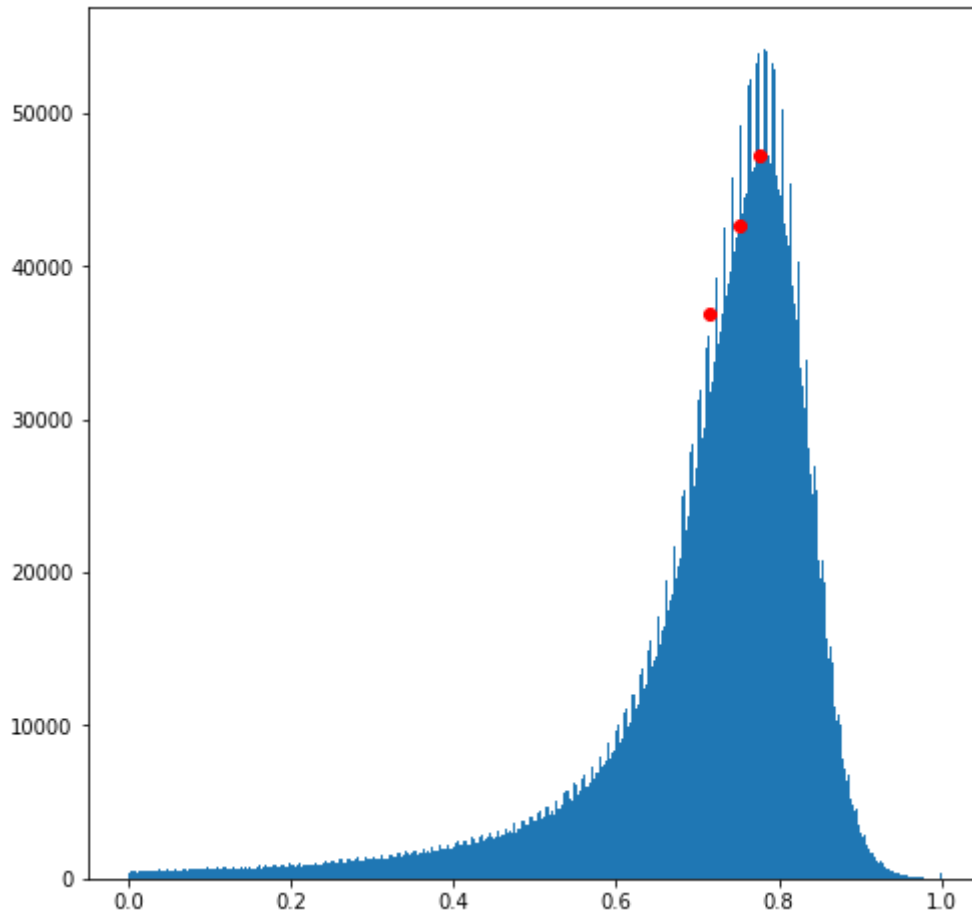
```
greater_bin: 0.7761750817298889  
index_bin: 631  
y: 47236.0
```



Mostrando a posição dos valores de média, mediana e moda, nessa ordem, no gráfico (pontos em vermelho).

In [24]:

```
plt.figure( figsize=(8, 8) )  
plt.hist( values, bins='auto' )  
plt.scatter(  
    [mean_value, median_value, mode_value],  
    [y_mean, y_median, y_mode],  
    # color = ['orange', 'pink', 'red'],  
    color = 'red',  
    zorder = 2 )  
plt.show()
```



Para exportar para PDF.

In [25]:

```
%%time
```

```
!jupyter nbconvert --to html ./T1_DanielCosta_TC.ipynb
```

```
[NbConvertApp] Converting notebook ./T1_DanielCosta_TC.ipynb to html
```

```
[NbConvertApp] Writing 982552 bytes to ./T1_DanielCosta_TC.html
```

```
CPU times: user 23.6 ms, sys: 9.08 ms, total: 32.7 ms
```

```
Wall time: 1.43 s
```