In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
from mylib import *
from skimage.segmentation import mark_boundaries, slic
from skimage.measure import label, regionprops, regionprops_table
from copy import deepcopy
%matplotlib notebook
```

In [2]:

```
secao = np.load('secao_do_plug.npy')
```

In [3]:

```
#secao = secao[::2,::2]
#sem reduzir a secao, as borda ficam mais suaves.
```

In [4]:

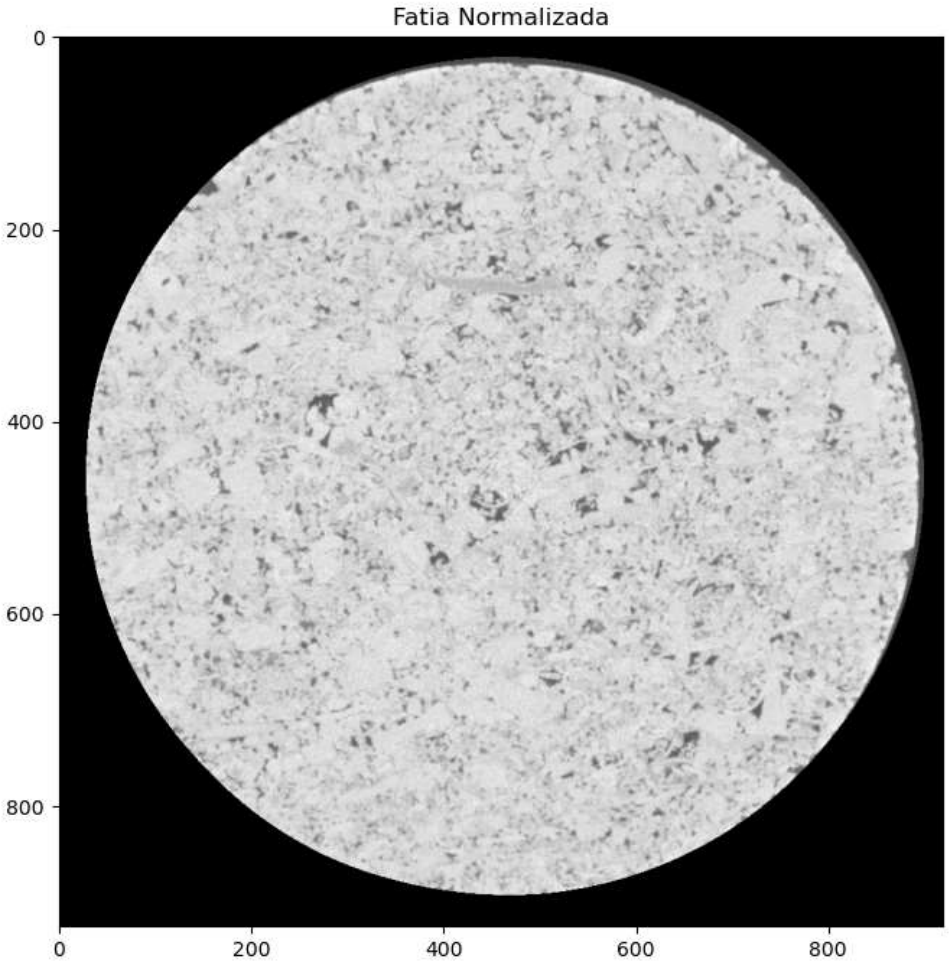
```
mascara = secao > 0
```

In [5]:

```
secao = secao.astype(float)
vmax = np.amax(secao)
vmin = np.amin(secao)
secao = (secao - vmin)/(vmax - vmin)
```

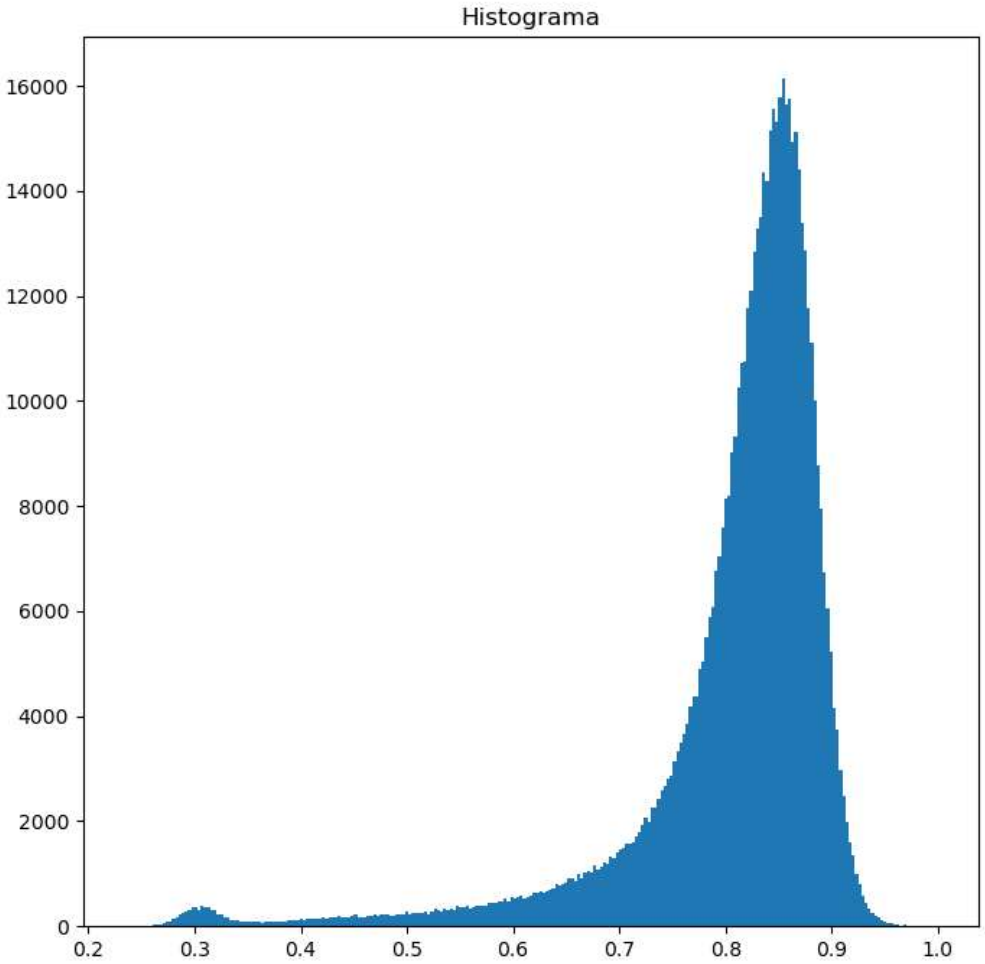
In [6]:

```
show_gray(secao, "Fatia Normalizada")
```


Fatia Normalizada
0
200
400
600
800
0
200
400
600
800

In [7]:

```
show_hist(secao[mascara].ravel(), "Histograma")
```



In [8]:

```
n_sp = 5000
```

In [9]:

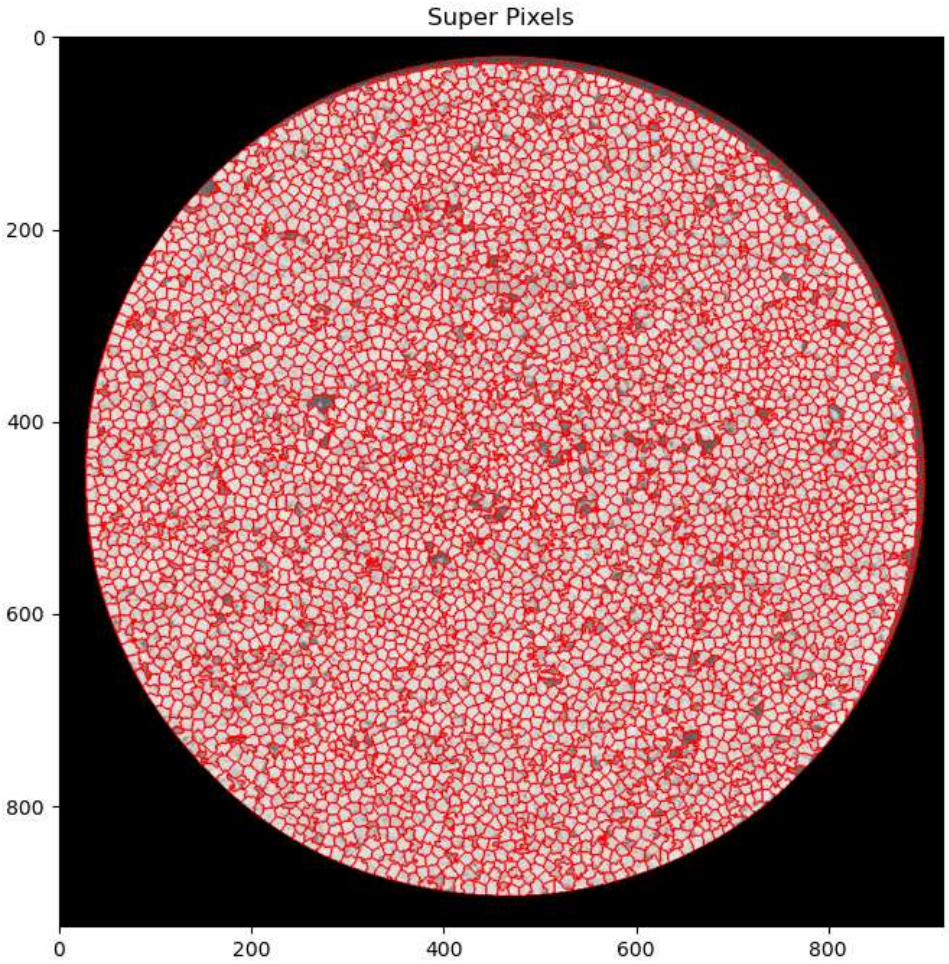
```
segmentos = slic(secao, n_segments=n_sp, compactness=0.06, slic_zero=False, mask=mascar
a)
```

In [10]:

```
img_sp = mark_boundaries(secao, segmentos, (1,0,0))
```

In [11]:

```
show_rgb(img_sp, "Super Pixels")
```



In [12]:

```
label_img = label(segmentos)
regions = regionprops(label_img, intensity_image=secao)
```

In [13]:

```
ys = []
xs = []
for props in regions:
    centroide = props.centroid
    ys.append(centroide[0])
    xs.append(centroide[1])
```

In [14]:

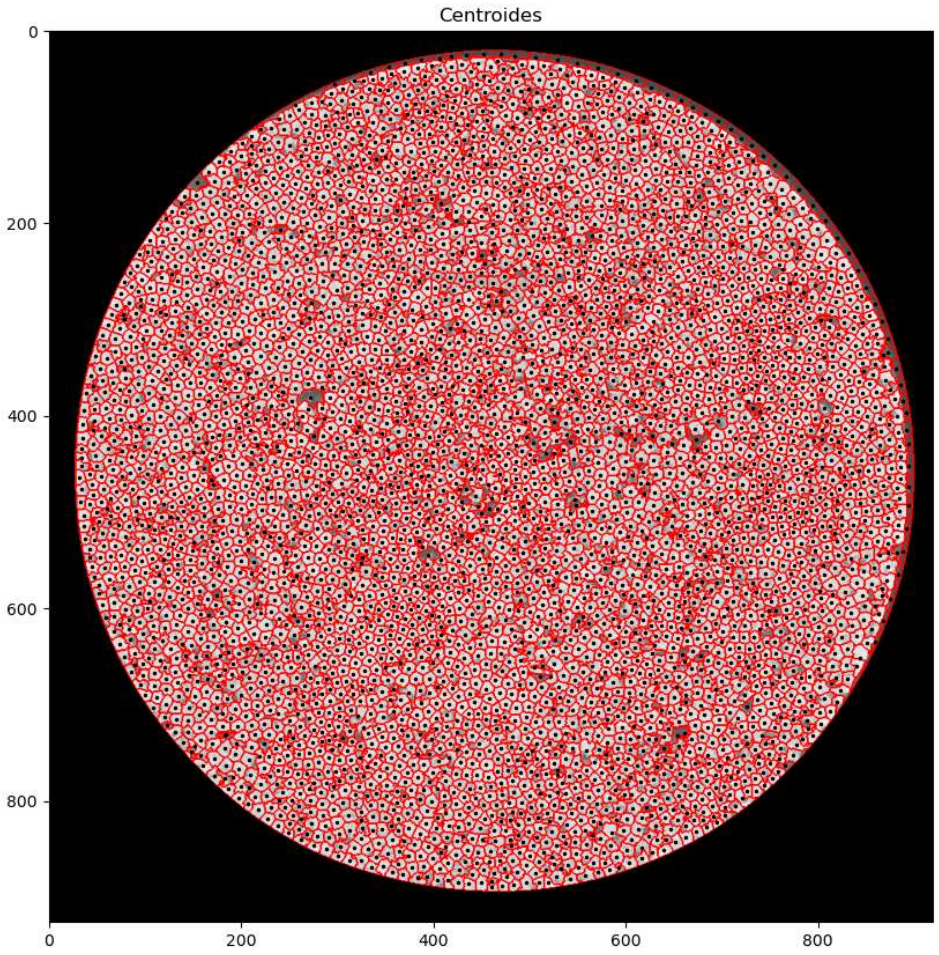
```
def show_centroids(img, title):
    fig, ax = plt.subplots(figsize=(10,10))
    ax.imshow(img, cmap=plt.cm.gray)

    for i in range(len(ys)):
        ax.plot(xs[i], ys[i], '.k', markersize=3)

    ax.axis((0, secao.shape[1], secao.shape[0], 0))
    plt.title(title)
    plt.show()
```

In [15]:

```
show_centroids(img_sp, "Centroides")
```



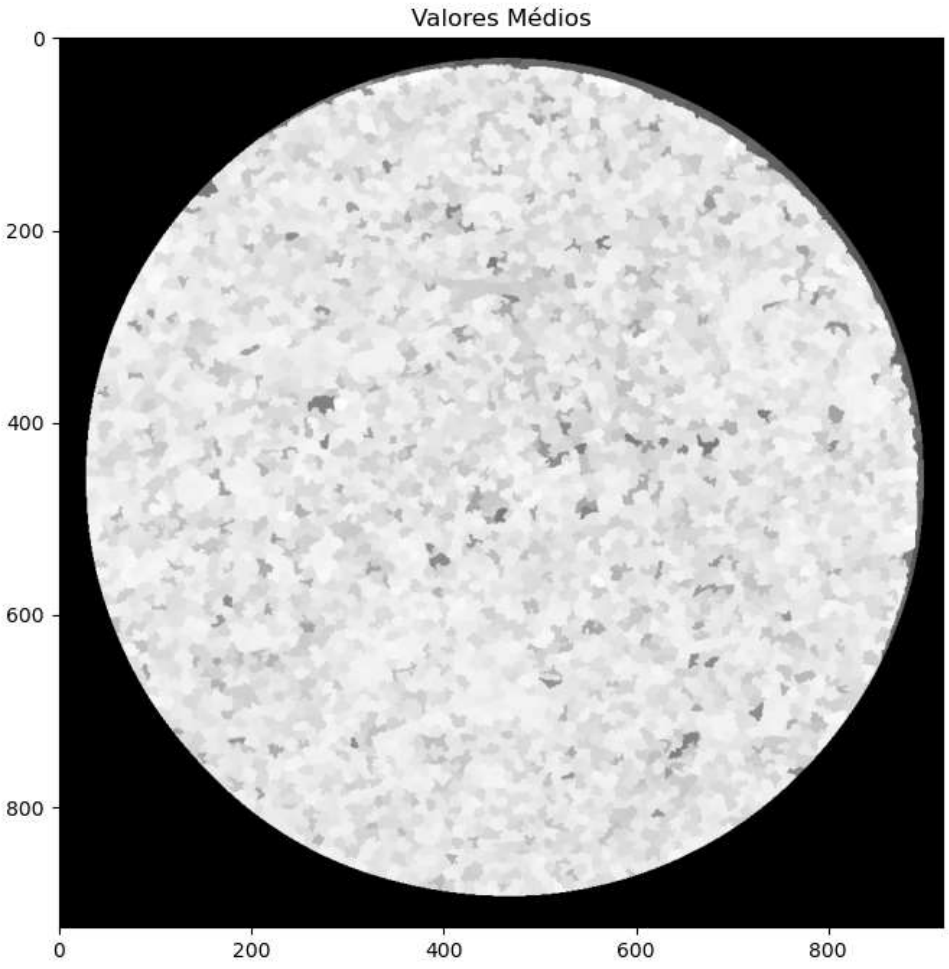
In [16]:

```
img_mean = deepcopy(secao)

for props in regions:
    mean = props.mean_intensity;
    coords = props.coords
    for i in range(coords.shape[0]):
        img_mean[coords[i][0]][coords[i][1]] = mean
```

In [17]:

```
show_gray(img_mean, "Valores Médios")
```

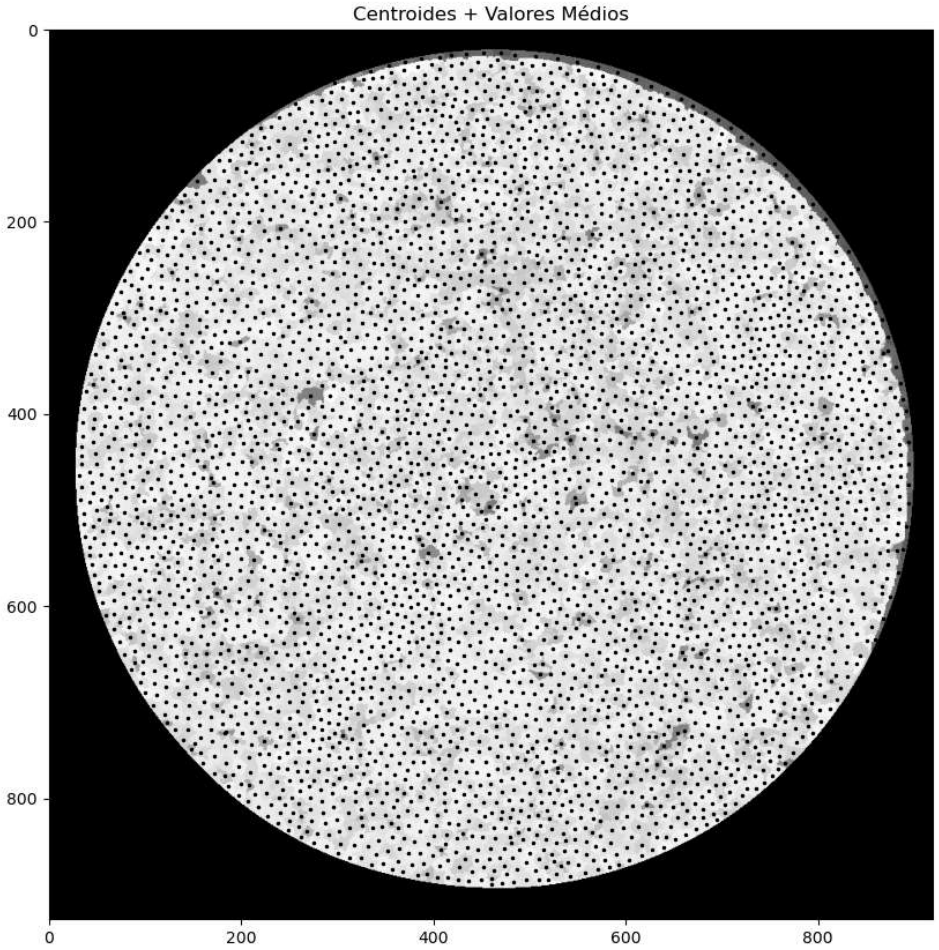


In [18]:

```
img_sp_mean = mark_boundaries(img_mean, segmentos, (1,0,0))
```

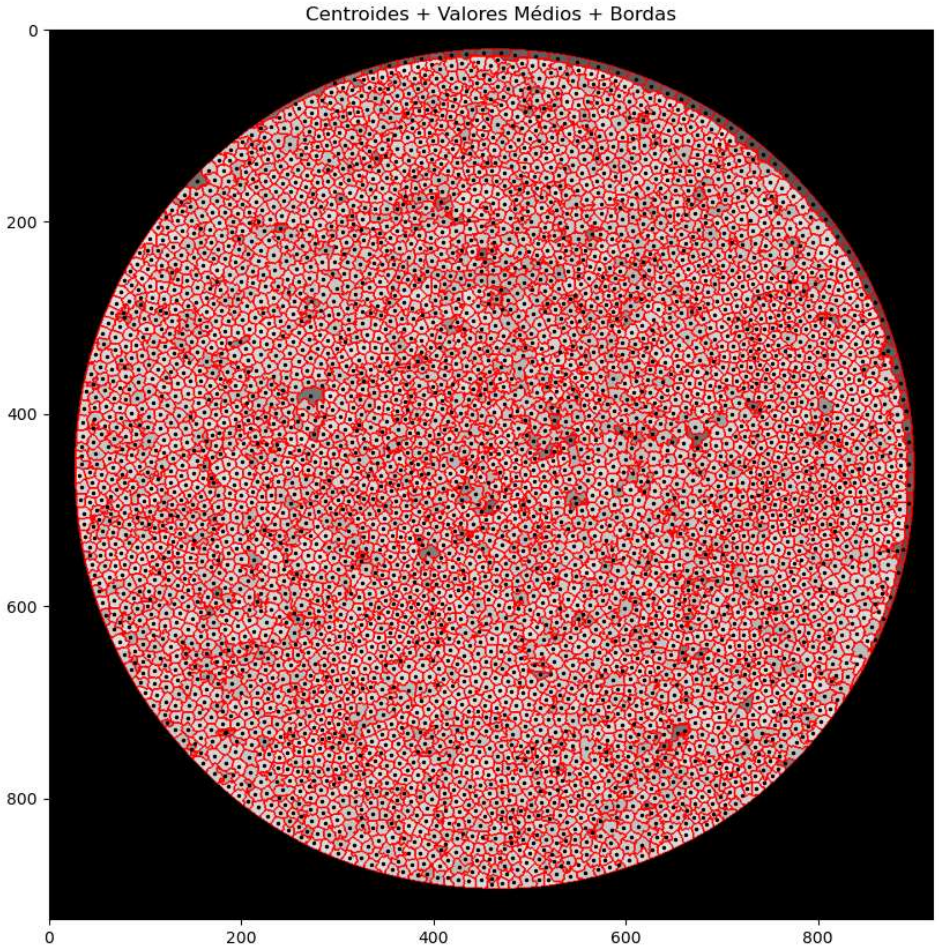
In [19]:

```
show_centroids(img_mean, "Centroides + Valores Médios")
```



In [20]:

```
show_centroids(img_sp_mean, "Centroides + Valores Médios + Bordas")
```



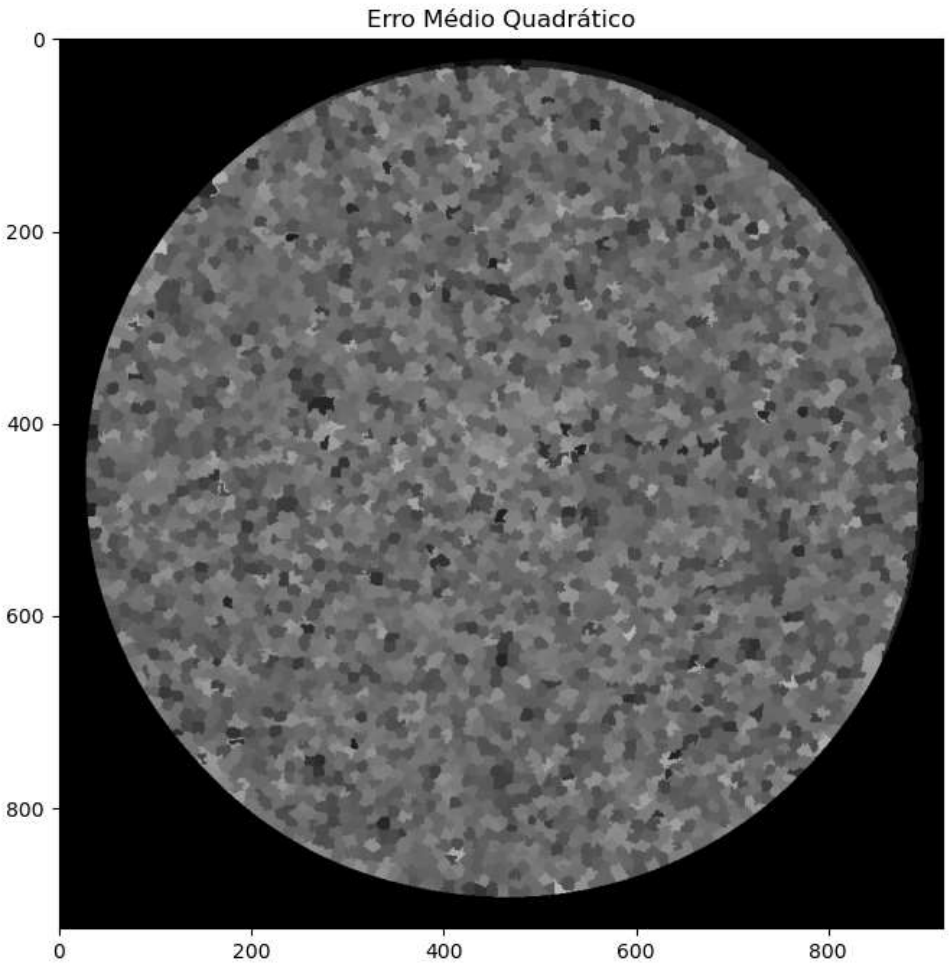
In [21]:

```
img_error = deepcopy(secao)

for props in regions:
    mean = props.mean_intensity
    intensity = props.intensity_image
    coords = props.coords
    error = np.square(intensity - mean).mean()
    for i in range(coords.shape[0]):
        img_error[coords[i][0]][coords[i][1]] = error
```

In [22]:

```
show_gray(img_error, "Erro Médio Quadrático")
```



In [23]:

```
print("Erro médio absoluto:", np.abs(img_mean - secao).mean())
print("Erro médio quadrático:", np.square(img_mean - secao).mean())
```

```
Erro médio absoluto: 0.025144023751576997
Erro médio quadrático: 0.001988332915214989
```