

# Tarefa 2 - Imagens Médicas 2

July 29, 2021

---

Universidade Federal de Uberlândia

Faculdade de Engenharia Elétrica

Imagens Médicas 2

Discente: Levy Gabriel da S. G.

---

```
[12]: from IPython.display import Image

def imageNcaption(filename, caption):
    display(Image(filename=filename))
    print(caption)
```

## 1 Tarefa 2 - Introdução ao ImageJ

### 1.1 GNU Octave e ImageJ

Antes de dar continuidade à atividade introdutória ao ImageJ, torna-se fundamental esclarecer ambos os *softwares* que serão utilizados ao longo desta disciplina de Imagens Médicas 2. O primeiro representado pela figura 1 é o GNU Octave, que fornecerá um nível de abstração de código ao nível do MATLAB e permitirá a execução de rotinas customizadas para processamento de imagens. O segundo na figura 2 é o *software* ImageJ construído em Java e que permitirá a visualização e manipulação de imagens com *built-in plugins* ou *plug-ins* customizáveis com facilidade e praticidade.

```
[13]: imageNcaption('octave.png', 'Figura 1 - Tela inicial do GNU Octave.')
```

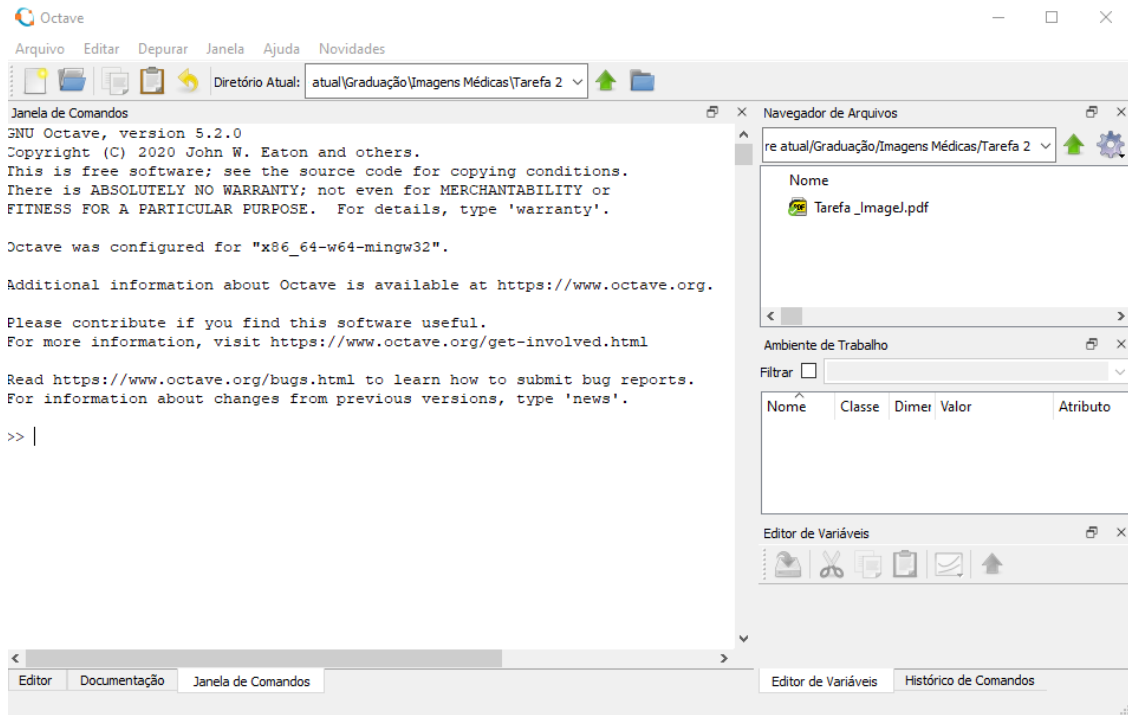


Figura 1 - Tela inicial do GNU Octave.

```
[14]: imageNcaption('imagej.png', 'Figura 2 - Tela inicial do ImageJ.')
```

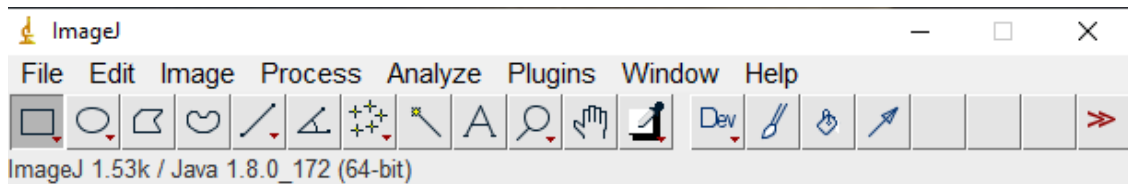


Figura 2 - Tela inicial do ImageJ.

Após isso será executada uma tarefa introdutória no ImageJ para familiarização da ferramenta.

## 1.2 Passo-a-passo com o ImageJ

Como o tutorial deixou em aberto a escolha da imagem a ser utilizada, os passos descritos serão, então, realizados com a imagem amostra do ImageJ chamada *Particles.gif*, como pode ser observada abaixo no contexto do programa.

```
[15]: imageNcaption('passo1.png', 'Figura 3 - Imagem Particles escolhida para análise.
→')
```

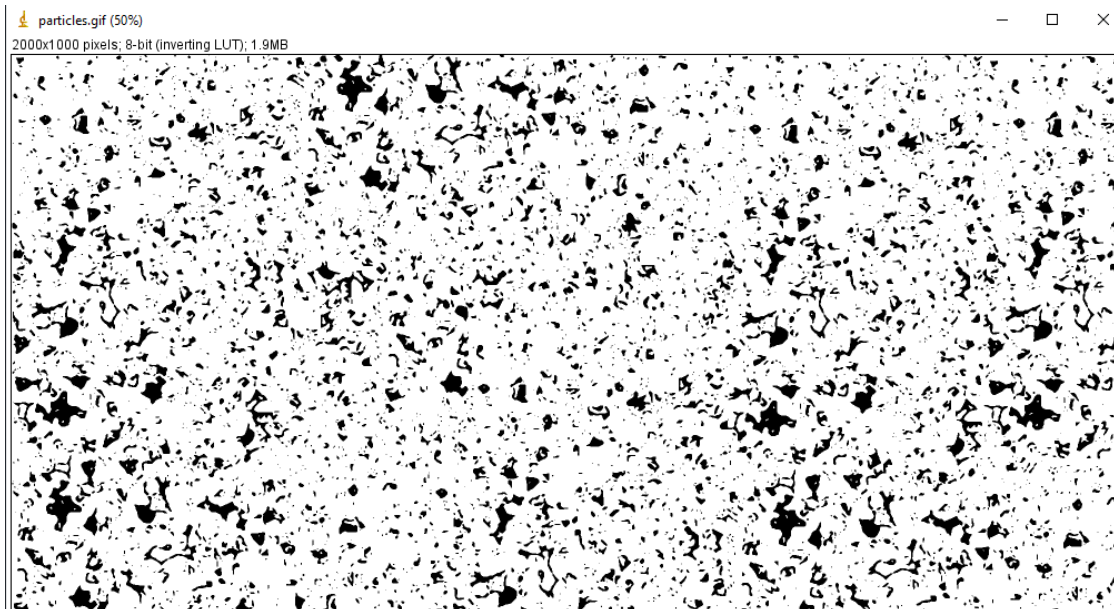


Figura 3 - Imagem Particles escolhida para análise.

Ao analisar a ferramenta de zoom, observam-se os limites de zoom para a presente imagem. Quando explorado em seu mínimo, o zoom chega a 1.4%. Quando explorado em seu máximo, este alcança 3200%. A figura 4 e a figura 5 ilustra ambas as situações.

```
[16]: imageNcaption('passo2.png', 'Figura 4 - Zoom mínimo.')
```

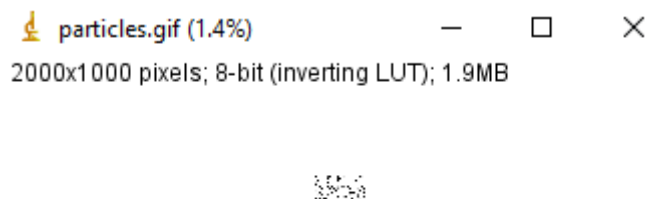


Figura 4 - Zoom mínimo.

```
[17]: imageNcaption('passo3.png', 'Figura 5 - Zoom máximo.')
```

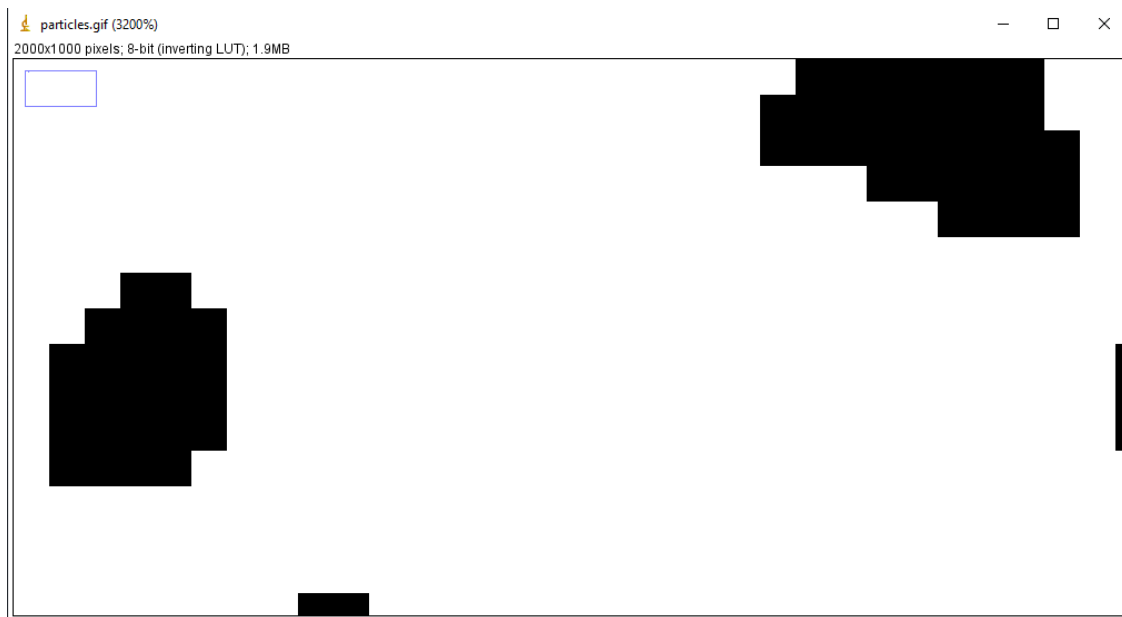


Figura 5 - Zoom máximo.

Ao avaliar a imagem em suas coordenadas de pixel  $100 \times 100$  e  $200 \times 200$  por meio da ferramenta *Pixel Inspection Tool*, porém pela natureza binária da imagem de cores somente branco ou preta observa-se que ambos os pixels são totalmente brancos, assim com nível 0. O exemplo é ilustrado na figura 6 e 7.

```
[18]: imageNcaption('passo4.png', 'Figura 6 - Avaliando o pixel 100x100.')
```

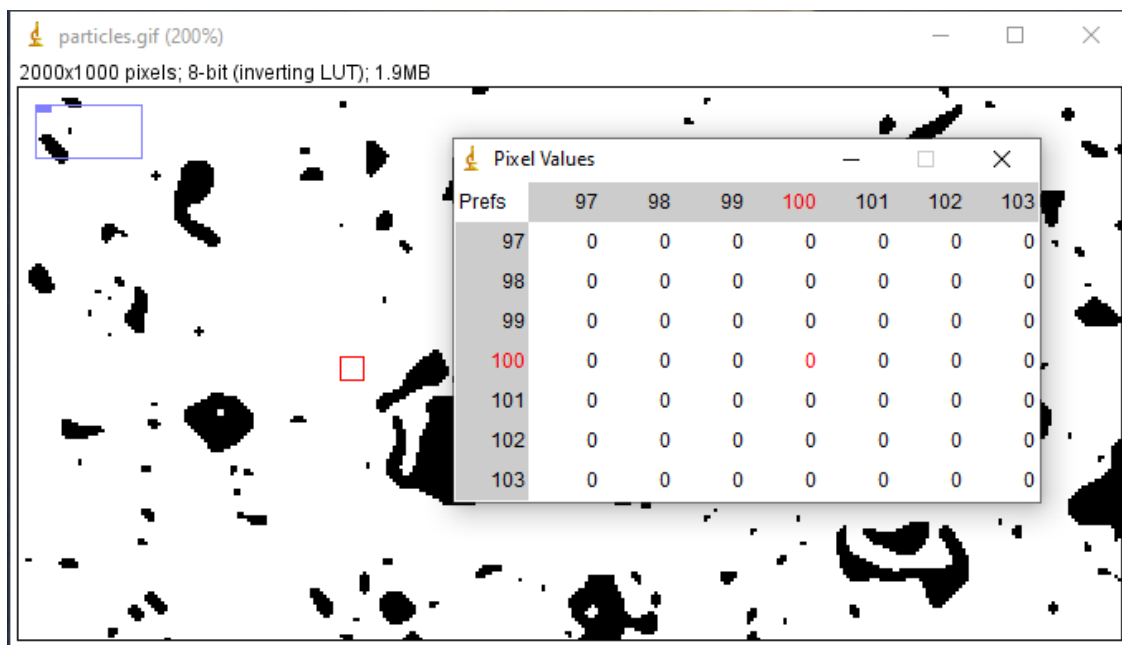


Figura 6 - Avaliando o pixel 100x100.

```
[19]: imageNcaption('passo5.png', 'Figura 7 - Avaliando o pixel 200x200.')
```

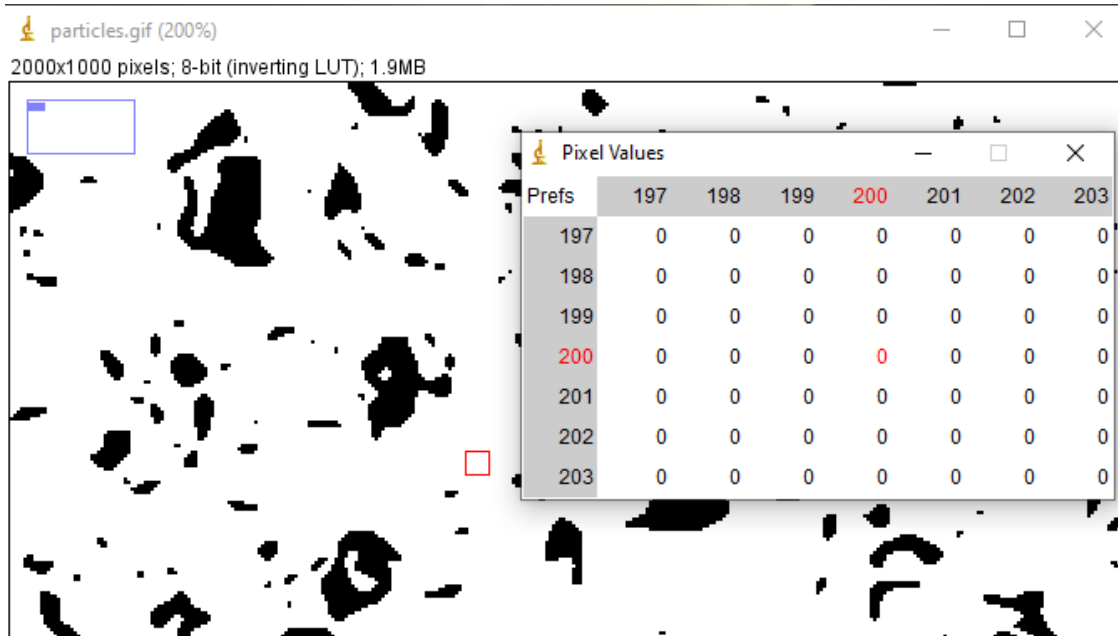


Figura 7 - Avaliando o pixel 200x200.

Para ilustrar uma região da imagem que em uma posição aleatória, cuja intensidade dos pixels seja diferente da encontrada anterior, como por exemplo a posição  $542 \times 243$ , que possui nível 255 e toda a vizinhança também como 255. A figura 8 ilustra os resultados.

```
[20]: imageNcaption('passo6.png', 'Figura 8 - Avaliando o pixel 542x243.')
```

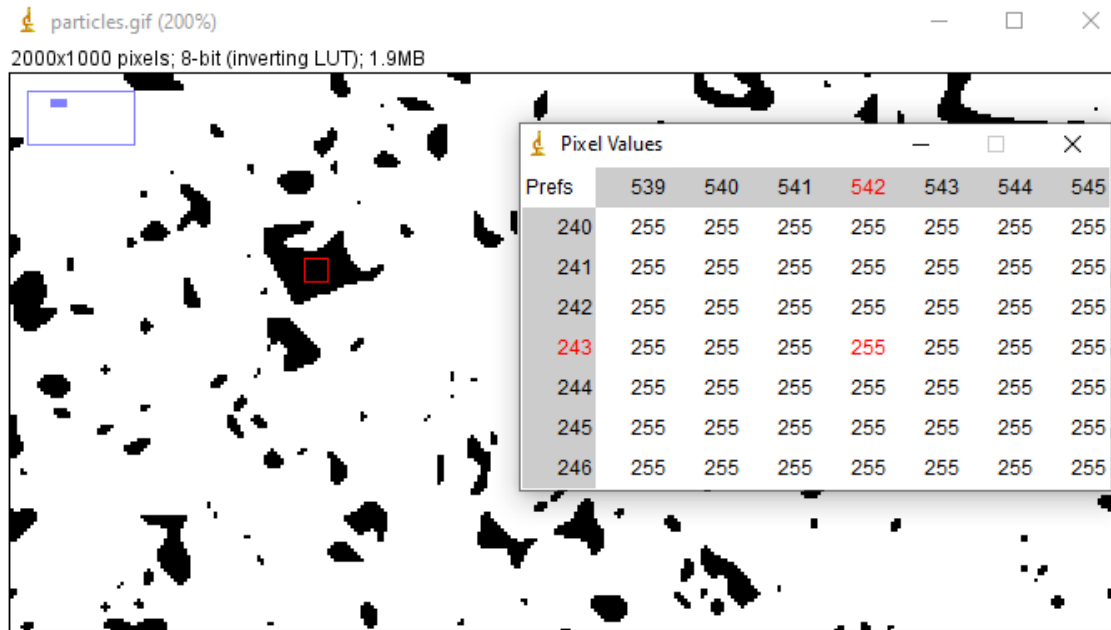


Figura 8 - Avaliando o pixel 542x243.

O imageJ também fornece informações sumárias sobre a imagem em sua barra superior, tais como dimensões, número de bits e o seu tamanho ocupado em memória. Ao observar a figura 9, observa-se que estes dados são preenchidos por:

- Largura de 2000 pixels;
- Altura de 1000 pixels;
- 8-bits de quantização;
- 1.9MB de espaço ocupado em memória.

```
[21]: imageNcaption('passo7.png', 'Figura 9 - Informações sumárias sobre a imagem.')
```

particles.gif (200%)  
2000x1000 pixels; 8-bit (inverting LUT); 1.9MB

Figura 9 - Informações sumárias sobre a imagem.

Ao mover o mouse sobre a imagem, observa-se a orientação com que a posição das imagens varia. Por exemplo, no eixo  $x$ , os valores da posição crescem à medida que avança-se à direita da imagem. No eixo  $y$  os valores da posição crescem de acordo com que avança-se para baixo na imagem. Assim as coordenadas  $x$  variando da esquerda para a direita e as coordenadas  $y$  variando de cima para baixo.

Nesse caso a coordenada  $(0, 0)$  situa-se na extremidade superior esquerda da imagem e a coordenada máxima ( $largura, altura$ ) situa-se na extremidade inferior direita da imagem.

O tutorial orienta inverter o eixo y para observar os resultados. A figura 10 ilustra a situação em que o eixo y foi invertido. A imagem foi posicionada em sua extremidade inferior direita e ao invés das coordenadas estarem perto do valor (*largura*, *altura*), elas na verdade estão próximas de (*largura*, 0) devido a inversão do eixo y.

```
[22]: imageNcaption('passo8.png', 'Figura 10 - Inversão do eixo y.')
```

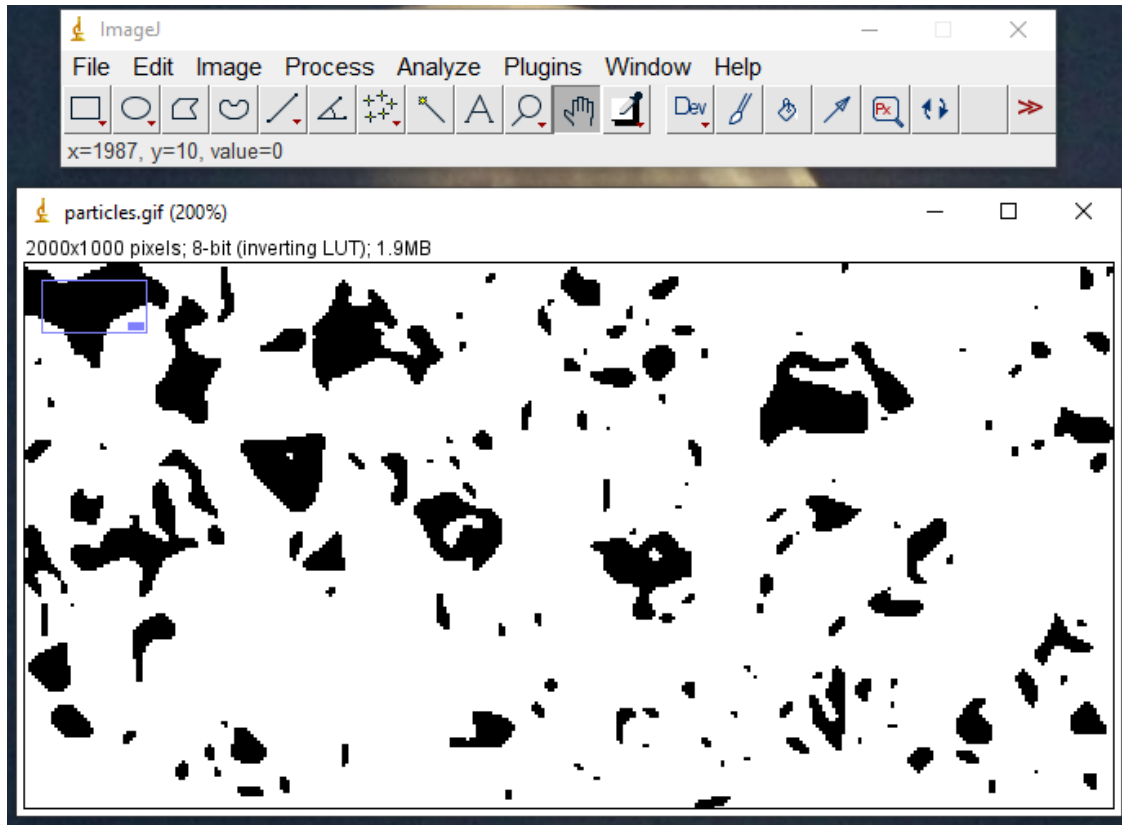


Figura 10 - Inversão do eixo y.