

### Suporte para GitHub:

Carregando arquivos pelo navegador: [https://drive.google.com/open?id=1Klf0HCJcB\\_4Q5B7efMXrs\\_YYDXGwH89U](https://drive.google.com/open?id=1Klf0HCJcB_4Q5B7efMXrs_YYDXGwH89U)  
Tutorial Basico GitHub com Eclipse e EGit Usando Chave SSH: <http://www.youtube.com/watch?v=fFBsazTSGZw>  
Usando Github com Github Desktop em Projetos Eclipse: <http://www.youtube.com/watch?v=EgHljYyS4U>  
Usando Github com SSH no Terminal Linux com chave gerada no Eclipse: <http://www.youtube.com/watch?v=0s699q5Sja4>  
Usando Github com SSH no Terminal Windows com chave gerada no Eclipse: <http://www.youtube.com/watch?v=DaydwPB2WSI>

### Gerando link compartilháveis:

<https://drive.google.com/file/d/1cyoxa5W67MY5xDM6gCYpklel1GU3QCKa>

### Vídeos Suporte:

#### Bubble Sort:

Teste de Mesa: <https://youtu.be/gLBIZYfBmsc>

Java: <https://youtu.be/il8umYPZa7w>

#### Merge Sort:

Teste de Mesa: <https://youtu.be/fw1dJOREzjY>

Java: <https://youtu.be/2-5eHsE4wdk>

**Para todos os exercícios, quando solicitado teste de mesa, carregar a solução para um drive compartilhado e quando solicitado desenvolvimento, definir o que se pede e aplicar o código em Java e carregar a solução no Github.**

1. Considere o seguinte vetor:

74	20	74	87	81	16	25	99	44	58
----	----	----	----	----	----	----	----	----	----

Fazer:

- O teste de mesa para a aplicação de ordenação por Bubble Sort, apresentando quantas rodadas são necessárias para que o vetor fique ordenado;
- O teste de mesa para a aplicação de ordenação por Merge Sort, detalhando as divisões e as operações com os vetores auxiliares;

2. Considere o seguinte vetor:

44	43	42	41	40	39	38
----	----	----	----	----	----	----

Fazer:

- O teste de mesa para a aplicação de ordenação por Bubble Sort, apresentando quantas rodadas são necessárias para que o vetor fique ordenado;
- O teste de mesa para a aplicação de ordenação por Merge Sort, detalhando as divisões e as operações com os vetores auxiliares;

3. Criar, em Java, uma Biblioteca que implemente as funcionalidades de ordenação do BubbleSort e uma Biblioteca que implemente as funcionalidades de ordenação do MergeSort.

4. Criar um projeto Java que receba as bibliotecas criadas no exercício 3 e apresente os resultados dos exercícios 1 e 2.