Estrutura de Dados

# Primeira versão do projeto da disciplina

## Comparação entre os algoritmos de ordenação elementar

Carlos Vinícius Nascimento Lira - Matrícula: 142088022

Dannilo Egito de Andrade - Matrícula: 161085245

Josênelle Cavalcante Santo - Matrícula: 152085408

# 1. Introdução

Este relatório corresponde ao relato dos resultados obtidos no projeto da disciplina de Estruturas de Dados (EDA) do curso de Ciências da Computação da Universidade Estadual da Paraíba - UEPB, que tem como objetivo principal, utilizar os dados da COVID-19 do Brasil.IO para estudar o desempenho dos algoritmos de ordenação.

O projeto consiste em preparar o dataset, que será um registro histórico de ocorrência de COVID-19 para todas as cidades e estados do Brasil. A planilha foi baixada localmente para poder processá-la em código Java. Só foram utilizados para esse projeto os dados mais atuais, os valores mais antigos ou repetidos foram desprezados, após feito esse filtro com os dados foi gerado um arquivo CSV,  para ser utilizado como data base.

Na segunda fase do projeto foi feita a implementação de todos os algoritmos de ordenação estudados (Selection Sort, Insertion Sort, Merge Sort, Quick Sort, Counting Sort, e HeapSort) para ordenar os registros de acordo com os seguintes parâmetros:

- Ordenação crescente por quantidade acumulada de óbitos;
- Ordenação crescente por quantidade acumulada de casos;
- Ordenação crescente por ordem alfabética pelo nome das cidades.

Na última fase cada arquivo de saída de ordenação é gerado com base no método de ordenação e nos elementos que devem ser ordenados. Como exemplo, do Quick Sort, 3 arquivos: qSort_ordena_obitos.csv e qSort_ordena_casos.csv, qSort_ordena_cidades.csv.

Em seguida, apresentaremos as seguintes seções: será descrito o método utilizado e o ambiente utilizado (processamento, memória, sistema operacional), apresentaremos os resultados obtidos (com gráficos e saída de execução dos algoritmos) e análise dos mesmo.

O repositório com o projeto e a análise estão disponíveis em: https://github.com/daniloegito/Projeto-EDA_DadosCovid

# 2. Descrição geral sobre o método utilizado

Para se obter todos os resultados foram realizados alguns testes, para cada um dos algoritmos implementados é analisando no melhor caso e no caso médio (para este projeto os testes de pior caso não foram implementados) de cada um dos algoritmos utilizados, levando em consideração o tempo em milisegundos de execução para cada teste analisado.

Esses testes foram feitos utilizando o compilador Eclipse IDE for Java, e foi levando em consideração o tempo em milissegundos que cada algoritmo utiliza para mostrar os resultados desejados.

Para as planilhas, tabelas e gráficos aqui apresentados foi utilizada a ferramenta Microsoft Word.

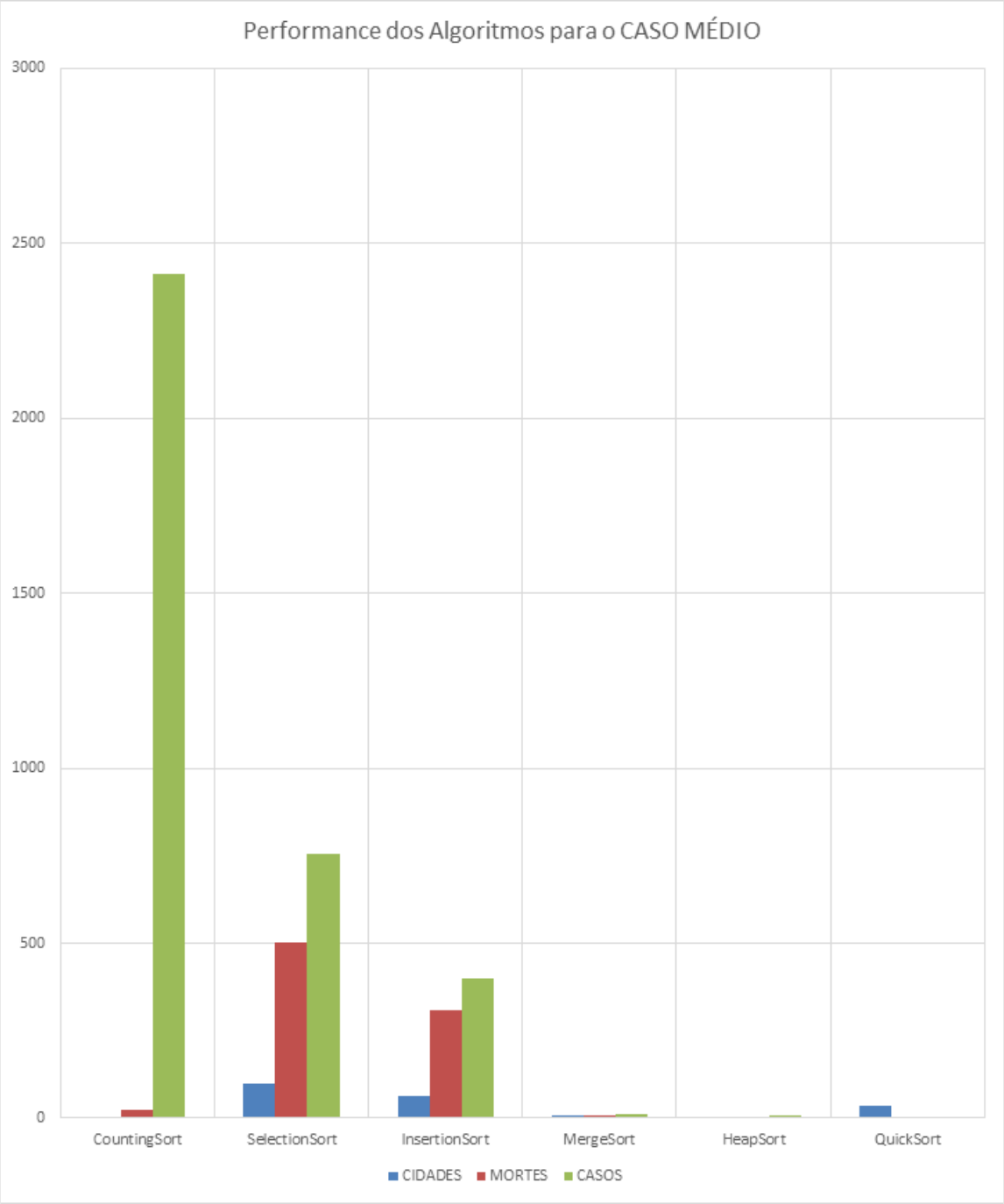
## Descrição geral do ambiente de testes

Os testes dos algoritmos deste projeto foram realizados utilizando os seguintes hardware e software:

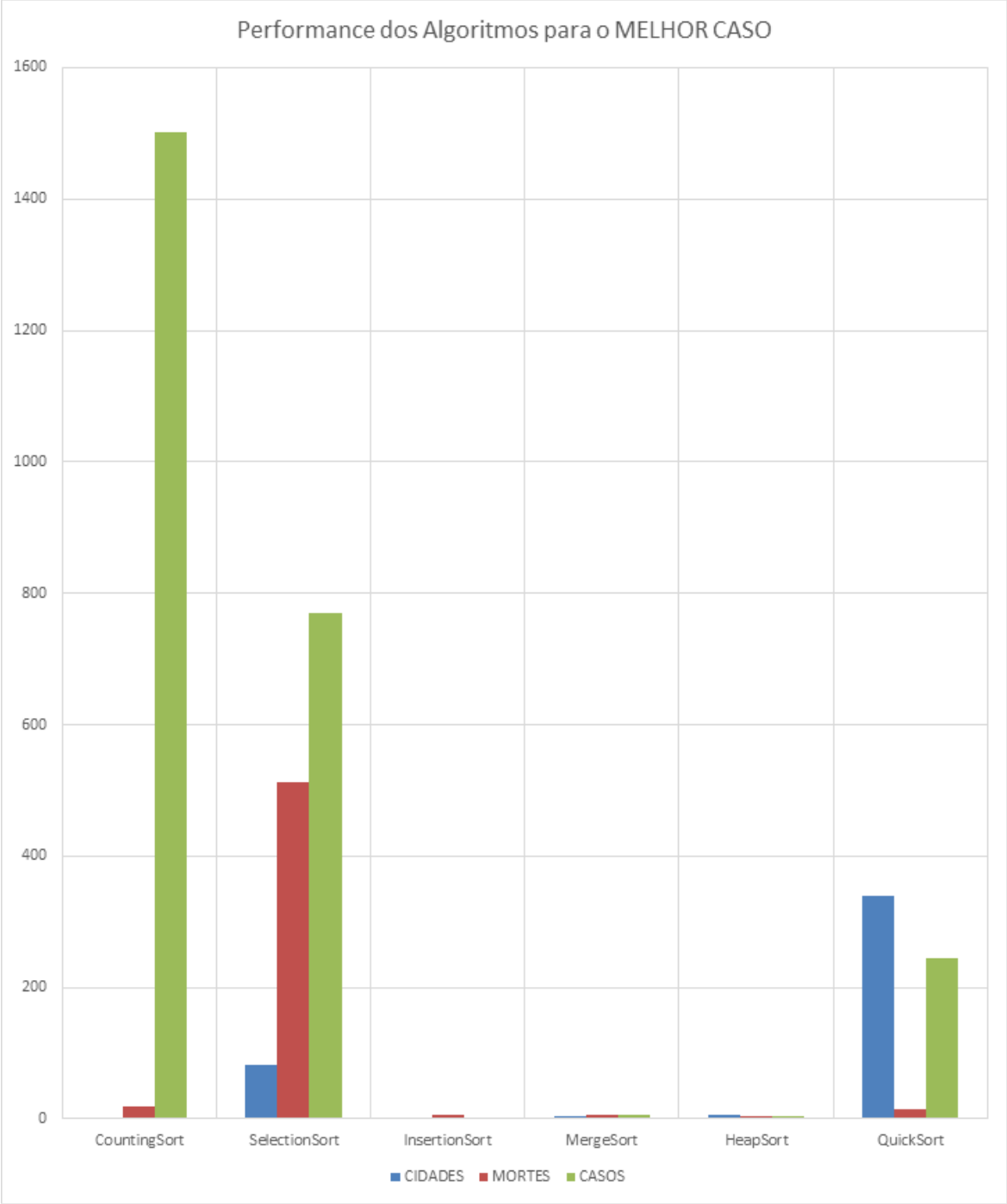
- Laptop ASUS Aspire - Intel Core I5 - 8G RAM - Windows 10
- Desktop - AMD Ryzen 5 1600 six core - 8GB RAM - Windows 10
- Laptop Lenovo Ideapad - Intel Core i3 - 4G RAM - Windows 10
- Laptop LG Ultra - Intel Celeron 4100 - 4GB RAM - Zorin OS (kernel linux)

# 3. Resultados e Análise

Os resultados obtidos pelos testes dos algoritmos representados por meio dos gráficos abaixo:



Como se pode observar no gráfico, os algoritmos MergeSort (cidades - 7ms) e HeapSort(cidades - 4ms) tiveram a melhor performance no caso médio. Já o SelectionSort (mortes - 755ms) e o CountingSort (casos - 2414ms), foram os algoritmos que tiveram a pior performance de tempo.



Como podemos observar no gráfico para o melhor caso, os algoritmos InsertionSort (casos - 1ms), MergeSort (casos - 7ms) e HeapSort (casos - 5ms) tiveram o melhor desempenho em milissegundos, já CountingSort (casos - 1502ms), o Selection Sort (mortes - 513ms) e o QuickSort(cidades - 339ms) tiveram os piores resultados em sua execução.

**Tabela comparativa referente a ordenação com entrada desordenada**

**(CASO MÉDIO)**

| | CASOS | MORTES | CIDADES |
|---|---|---|---|
| Melhor resultado – > | QuickSort | QuickSort | HeapSort |
| Médio resultado – > | HeapSort | HeapSort | MergeSort |
| Resultado Médio – > | MergeSort | MergeSort | QuickSort |
| Resultado Médio – > | InsertionSort | CountingSort | InsertionSort |
| **Pior resultado – >** | SelectionSort | InsertionSort | SelectionSort |
| **Pior resultado – >** | CountingSort | SelectionSort | -------- |

De acordo com os resultados obtidos pela tabela de caso médio, o algoritmo QuickSort se destaca como melhor performance em dois quesitos em casos de Covid-19 e de mortes, com tempo de execução em ambos os testes de 5ms. Já no quesito Cidades o algoritmo HeapSort tem melhor desempenho, com tempo de execução de 4ms.

Já quando levamos em consideração quem teve os piores resultados na tabela de caso médio, podemos destacar SelectionSort em duas ocasiões em Casos de Covid-19 e em cidades, com tempo de execução respectivamente, 755ms e 93ms, o InsertionSort teve o pior desempenho em mortes, com tempo de execução 63ms e SelectionSort com 93ms. O CountingSort é considerado o pior caso em mortes, com 2414ms de tempo de execução.

**Tabela referente a ordenação com entrada ordenada (MELHOR CASO)**

| | CASOS | MORTES | CIDADES |
|---|---|---|---|
| Melhor resultado – > | InsertionSort | HeapSort | InsertionSort |
| Médio resultado – > | HeapSort | InsertionSort | HeapSort |
| Resultado Médio – > | MergeSort | MergeSort | MergeSort |
| Resultado Médio – > | QuickSort | QuickSort | SelectionSort |
| **Médio caso – >** | SelectionSort | CountingSort | QuickSort |
| **Pior caso – >** | CountingSort | SelectionSort | ---------- |

De acordo com os resultados obtidos pela tabela de Melhor Caso, o algoritmo InsertionSort com tempo de execução de 1ms, se destaca como melhor performance em casos de Covid_19 e cidades  e HeapSort tem melhor performance em número de Mortos com tempo de execução de  5ms.

Já quando levamos em consideração quem teve os piores resultados na tabela de Melhor Caso, o algoritmo CountingSort em número de casos de Covid-19 e o SelectionSort em número de Mortes foram os piores, com tempo de execução, respectivamente de 1502ms e 513ms.

**Ao executar a aplicação, é gerado um log de saída com o tempo em milissegundos que cada algoritmo precisou para realizar a ordenação. Como este:**

----- ALGORITMOS DE ORDENAÇÃO E TEMPO GASTO -------

ALGORITMO SELECTION SORT

CASOS

Tempo de execução - caso médio: 755ms

Tempo de execução - melhor caso: 769ms

MORTES

Tempo de execução - caso médio: 502ms

Tempo de execução - melhor caso: 513ms

CIDADES

Tempo de execução - caso médio: 97ms

Tempo de execução - melhor caso: 83ms

ALGORITMO INSERTION SORT

CASOS

Tempo de execução - caso médio: 399ms

Tempo de execução - melhor caso: 1ms

MORTES

Tempo de execução - caso médio: 307ms

Tempo de execução - melhor caso: 7ms

CIDADES

Tempo de execução - caso médio: 63ms

Tempo de execução - melhor caso: 1ms

ALGORITMO MERGE SORT

CASOS

Tempo de execução - caso médio: 12ms

Tempo de execução - melhor caso: 7ms

MORTES

Tempo de execução - caso médio: 9ms

Tempo de execução - melhor caso: 7ms

CIDADES

Tempo de execução - caso médio: 7ms

Tempo de execução - melhor caso: 5ms

ALGORITMO QUICK SORT

CASOS

Tempo de execução - caso médio: 5ms

Tempo de execução - melhor caso: 245ms

MORTES

Tempo de execução - caso médio: 5ms

Tempo de execução - melhor caso: 14ms

CIDADES

Tempo de execução - caso médio: 37ms

Tempo de execução - melhor caso: 339ms

ALGORITMO COUNTING SORT

CASOS

Tempo de execução - caso médio: 2414ms

Tempo de execução - melhor caso: 1502ms

MORTES

Tempo de execução - caso médio: 22ms

Tempo de execução - melhor caso: 19ms

ALGORITMO HEAP SORT

CASOS

Tempo de execução - caso médio: 8ms

Tempo de execução - melhor caso: 5ms

MORTES

Tempo de execução - caso médio: 5ms

Tempo de execução - melhor caso: 5ms

CIDADES

Tempo de execução - caso médio: 4ms

Tempo de execução - melhor caso: 6ms