



# Trabalho Prático III

---

## Regras Básicas

1. extends Trabalho Prático 02
2. Fique atento ao Charset dos arquivos de entrada e saída.
3. em caso de empate das chaves, favor desempatar pelo campo `nome`.
4. para os campos do tipo `String` é recomendado usar o método `compareTo` para ordenação.



Steam é uma plataforma criada em 2003 pela Valve que revolucionou o mundo dos games. O serviço apresentou recursos inovadores como o download de jogos em formato digital, o desbloqueio de conquistas (achievements) e a interação entre usuários. Atualmente, milhões de pessoas utilizam o serviço todos os dias. O download do programa é de graça, mas muitos games são pagos, embora a Steam ocasionalmente libere alguns jogos grátis durante finais de semana. O serviço possui hoje aplicativos para Android e iOS e versões para Mac e Linux. Além disso, a versão do Brasil conta com preços em reais, e você pode pedir reembolso de jogos que não rodaram ou que não tenham atendido às suas expectativas.

Neste Trabalho Prático sua tarefa é organizar as informações dos jogos disponíveis para exibição ao usuário. Entretanto, esses dados estão espalhados em um arquivo `.csv` obtidos a partir de uma busca na plataforma. Todos esses dados estão agrupados no arquivo `games.csv`, e o mesmo deve ser colocado na pasta `/tmp/`.<sup>1</sup> Para isso, você deve ler, organizar e armazenar os dados de cada jogo em memória, utilizando as estruturas de dados em aula (Lista, Pilhas, Filas, Árvores e Hash). Em seguida executar as operações descritas nos arquivos de entrada. Muito cuidado ao realizar o *parser* do texto. Fique atento a descrição dos dados que serão lidos e manipulados pelo seu sistema.

---

<sup>1</sup>Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta `/tmp/`.

# Algoritmos de Ordenação

1. **Ordenação por Seleção em Java:** Na classe `Lista`, implemente o algoritmo de ordenação por seleção considerando que a chave de pesquisa é o atributo **nome**. A entrada e a saída padrão são iguais às da primeira questão do Trabalho Prático II, contudo, a saída corresponde aos objetos ordenados. Além disso, crie um arquivo de log na pasta corrente com o nome `matricula_selecao.txt` com uma única linha contendo sua matrícula, número de comparações (entre elementos do *array*), número de movimentações (entre elementos do *array*) e o tempo de execução do algoritmo de ordenação. Todas as informações do arquivo de log devem ser separadas por uma tabulação `'\t'`.
2. **Ordenação por Seleção Recursiva em C:** Repita a questão anterior, contudo, usando a Seleção Recursiva. A entrada e a saída padrão serão iguais às da questão anterior. O nome do arquivo de log será `matricula_selecaoRecursiva.txt`.
3. **Ordenação por Inserção em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo de Inserção, fazendo com que a chave de pesquisa seja o atributo **app\_id**. O nome do arquivo de log será `matricula_insercao.txt`.
4. **Shellsort em C:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Shellsort, fazendo com que a chave de pesquisa seja o atributo **app\_id**. O nome do arquivo de log será `matricula_shellsort.txt`.
5. **Heapsort em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Heapsort, fazendo com que a chave de pesquisa seja o atributo **release\_date**. O nome do arquivo de log será `matricula_heapsort.txt`.
6. **Quicksort em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Quicksort, fazendo com que a chave de pesquisa seja o atributo **release\_date**. O nome do arquivo de log será `matricula_quicksort.txt`.
7. **Counting Sort em C:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Counting Sort, fazendo com que a chave de pesquisa seja o atributo **upvotes**. O nome do arquivo de log será `matricula_countingsort.txt`.
8. **Bolha em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo da Bolha, fazendo com que a chave de pesquisa seja o atributo **developers**. O nome do arquivo de log será `matricula_bolha.txt`.
9. **Mergesort em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Mergesort, fazendo com que a chave de pesquisa seja o atributo **upvotes**. O nome do arquivo de log será `matricula_mergesort.txt`.