



## GCC216 - Estruturas de Dados

### Projeto prático - Ordenação em memória secundária

**Estudantes Responsáveis:**

Pedro Henrique Cabral Moreira  
Esther Silva de Magalhães  
Ítalo Alves Rabelo

## Objetivo

O objetivo deste projeto prático é realizar a ordenação do arquivo csv "data\_athlete\_event.csv" utilizando-se métodos de ordenação externa.

## Descrição da estrutura do código

### - Métodos:

"divideArquivo()" - designado para dividir o arquivo "data\_athlete\_event.bin" em arquivos menores com 50.000 registros cada.

"intercala()" - alterna dois vetores utilizando o atributo "city" como chave de ordenação principal e "id" como chave secundária.

"ordenaArquivos()" - ordena os subarquivos a partir dos atributos "city" e "id" respectivamente.

"mesclaArquivos()" - método para reescrever os subarquivos ordenados no arquivo principal.

"mergeSortExterno()" - método que realiza a chamada das funções descritas acima para realizar a ordenação em memória secundária do arquivo binário principal.

## Descrição em alto nível do código

De modo geral, o projeto realiza a leitura da base de dados no formato csv e converte-o em arquivo binário. Para se ordenar grandes volumes de dados, é necessário dividir o arquivo principal em subarquivos menores, de forma a evitar a sobrecarga da memória principal durante a execução do programa.

Neste projeto, o método "divideArquivo()" foi utilizado para fraccionar o arquivo principal "data\_athlete\_event.csv" em arquivos com 50.000 registros cada.

Posteriormente, é realizada a ordenação dos arquivos menores a partir de dois atributos pré-definidos: City(char array), como chave principal e ID(integer), como chave secundária. A seguir, os registros são reescritos de forma ordenada no arquivo principal.

## **Conclusão**

No curso de realização deste trabalho, o grupo priorizou o versionamento de código com o fito de manter pleno controle sobre a funcionalidade e eficiência do projeto. Todavia, algumas dificuldades surgiram, como o tratamento de exceções, redução do tempo de leitura e escrita dos registros e manipulação do arquivo binário.

Como consideração final, destaca-se a importância de entender bem os conceitos teóricos e práticos apresentados durante as aulas de Estruturas de Dados para a realização deste trabalho.