



Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática
Algoritmos e Estruturas de Dados III

Trabalho Prático I
05 pontos

O trabalho deve ser feito **individualmente** ou em grupos de no **máximo 2 alunos**.

Data de entrega: 05/03/2023

Penalidade por atraso: a cada dia corrido de atraso, a nota será penalizada em 2 pontos.

Penalidade por cópia: trabalhos iguais não são aceitos (nota 0).

Etapas dos TPS:

- **Etapa 1: Criação da base de dados e Ordenação Externa: Implementação + Vídeo**
- Etapa 2: Indexação com Árvore B+, Hash e Lista Invertida: Implementação + Vídeo
- Etapa 3: Compactação com Huffman e LZW: Implementação + Vídeo
- Etapa 4: Casamento de Padrões: Implementação + Vídeo
- Etapa 5: Criptografia: Implementação + Vídeo + Relatório Final

Descrição do TP1:

Neste trabalho, você deverá selecionar uma base de dados que permita a representação das entidades em registros com campos de cada um dos tipos abaixo.

1. String de tamanho fixo
2. String de tamanho variável
3. Data
4. Lista de valores com separador a definir
5. Inteiro ou Float

Fique à vontade para selecionar uma base de dados. Sugerimos a seleção de uma base que faça sentido para você, isso pode fornecer um incentivo extra para um trabalho tão extenso e desafiador. Use fontes de bases de dados como o [Kaggle](https://www.kaggle.com/), ou outra que desejar, desde que cumpra os requisitos acima e seja de domínio público. Se tiver dúvida se determinada base de dados é válida, pergunte ao professor da disciplina.

Após selecionar a base de dados desejada, você deverá implementar um sistema responsável por realizar operações de CRUD (create, read, update e delete) em um arquivo sequencial. Como estas operações podem resultar na criação de espaços de memória que não podem ser aproveitados, você deverá então proceder com a elaboração de algoritmos para realizar a reordenação do arquivo.



Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática
Algoritmos e Estruturas de Dados III

Orientações:

- O sistema deve ser implementado em Java. Todo o código deve ser de autoria do grupo (com exceção para bibliotecas/classes relacionadas a aberturas e escritas/leituras de arquivos e conversões entre atributos e campos).
- Caso algum dos tipos acima citados não esteja presente na base de dados escolhida, você poderá estabelecer algum critério que permita a inclusão do campo. Exemplo: inclusão de ID de sequencial, inclusão da data/hora de carga.
- Todo o código deve ser comentado de modo a se compreender a lógica utilizada. A não observância deste critério implica na redução da nota final em 50%.
- A estrutura do arquivo, onde as operações de CRUD serão realizadas, deve ser a seguinte:
 - Deve-se utilizar um **int no cabeçalho** para armazenar o último valor de id utilizado.
 - Os registros do arquivo devem ser compostos por:
 - **Lápide** - Byte que indica se o registro é válido ou se é um registro excluído;
 - **Indicador de tamanho do registro** - Número inteiro que indica o tamanho do vetor de bytes;
 - **Vetor de bytes** - Bytes que descrevem o objeto.
- Os objetos utilizados devem possuir os atributos que representam a entidade da base de dados que você escolheu.

Parte 1 - CRUD:

O sistema deverá oferecer uma tela inicial (com uso pelo terminal) com um menu com as seguintes opções:

- Realizar a carga da base de dados selecionada, através da importação de arquivo CSV, de rota de API ou outro formato que julgar pertinente.
- Ler um registro (id) -> esse método deve receber um id como parâmetro, percorrer o arquivo e retornar os dados do id informado.



Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática
Algoritmos e Estruturas de Dados III

- Atualizar um registro -> esse método deve receber novas informações sobre um objeto e atualizar os valores dele no arquivo. Observe duas possibilidades que podem acontecer:
 - O registro mantém seu tamanho - Nenhum problema aqui. Basta atualizar os dados no próprio local.
 - O registro aumenta ou diminui de tamanho - O registro anterior deve ser apagado (por meio da marcação lápide) e o novo registro deve ser escrito no fim do arquivo.
- Deletar um registro (id) -> esse método deve receber um id como parâmetro, percorrer o arquivo e colocar uma marcação (lápide) no registro que será considerado deletado.

Parte 2 - ORDENAÇÃO EXTERNA:

Realizar a ordenação do arquivo. Para isso, deve ser realizada a implementação da ordenação externa, considerando a memória principal com limitação de “m” registros (parametrizável) e usando “n” caminhos (parametrizável). O sistema deverá oferecer, a partir da tela inicial, um submenu de Ordenação Sequencial com as seguintes opções de algoritmos de ordenação externa, por você implementados:

- Intercalação balanceada comum
- Intercalação balanceada com blocos de tamanho variável
- Intercalação balanceada com seleção por substituição

O que deve ser entregue:

Implementação

- Carga da Base de dados e CRUD
- Algoritmos de Ordenação Externa

Além da Implementação, o grupo deve criar um vídeo (duração máxima de 10 minutos), com:

- Explicação das principais decisões de implementação dos códigos criados.
- Demonstração da execução do sistema.
- Testes e resultados realizados

Critérios para avaliação

- Implementação do sistema (05 pontos)
 - Correção e robustez dos programas
 - Conformidade às especificações
 - Clareza de codificação
 - Critérios de escolha
- Vídeo (1 ponto)

NOTA FINAL = Implementação x Vídeo

Observação final: ponto(s) extra(s) pode(m) ser dado(s) para trabalhos considerados excelentes.