



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

## **Trabalho Prático I**

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

**Data de entrega:** 08/05/2025

**Valor:** 5 pontos

### **Etapas dos TPS:**

- Etapa 1: Criação da base de dados: Implementação
- **Etapa 2: Indexação: Implementação → 08/05**
- Etapa 3: Compactação: Implementação → 22/05
- Etapa 4: Casamento de Padrões: Implementação → 05/06
- Etapa 5: Criptografia: Implementação → 19/06

### **Descrição do TP1:**

Neste trabalho, você deverá manipular de forma indexada a base de dados criada no TP1.

### **Orientações:**

### **INDEXAÇÃO:**

- **Orientações para a criação do arquivo de índices usando Árvore B ou B+:**
  - O arquivo de índices deve usar a estrutura de Árvore B ou B+, usando como chave o campo *id*.
  - Você deve identificar e escolher qual árvore será usada (B ou B+). A escolha deve ser justificada.
  - A ordem da árvore deve ser parametrizada.
  - O arquivo de índices deve conter o *id* e a posição do registro (referente a esse *id*) no arquivo de dados.
  - Sempre que acontecerem alterações no arquivo de dados, novas alterações devem ser feitas no arquivo de índices, mantendo sempre a coerência entre esses arquivos.
  - O arquivo de índice criado deve possibilitar a realização de buscas no arquivo de dados.



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

- **Orientações para a criação do arquivo de índices usando Hashing Estendido:**
  - O arquivo de índices deve usar a estrutura de Hashing Estendido, usando como chave o campo *id*.
  - Você deve identificar e escolher em seu arquivo o campo que será indexado. Cada escolha deve ser justificada.
  - Deve-se usar uma função hash, tal função pode ser definida pelo grupo que justificará a escolha.
  - Cada bucket pode armazenar até X registros, sendo X definido pelo grupo que justificará a escolha.
  - O arquivo de índices deve conter o id e a posição do registro (referente a esse id) no arquivo de dados.
  - Sempre que acontecerem alterações no arquivo de dados, novas alterações devem ser feitas no arquivo de índices, mantendo sempre a coerência entre esses arquivos.
  - Deve existir a possibilidade de realizar buscas usando a estrutura de índices de Hashing Estendido.

**O que deve ser entregue:**

**Implementação**

- CRUD com apoio dos índices

Além da Implementação, o grupo deverá apresentar, máximo 5 minutos, o trabalho, com:

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

**Critérios para avaliação**



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

Critério	Atende perfeitamente (1,0)	Atende de forma satisfatória (0,8)	Atende parcialmente (0,6)	Atende de forma insatisfatória (0,3)	Não atende (0)
<b>Correção e robustez dos programas</b>	O sistema está livre de erros, falhas e lida bem com entradas inesperadas.	O sistema tem poucos erros ou falhas, mas lida corretamente com entradas esperadas e alguns edge cases.	O sistema falha em algumas condições ou não lida com todas as entradas inesperadas corretamente.	O sistema tem falhas que afetam o funcionamento básico e não lida bem com entradas inesperadas.	O sistema apresenta falhas graves que comprometem a execução e não lida com entradas adequadas.
<b>Conformidade às especificações</b>	A implementação segue fielmente todas as especificações do problema, sem falhas.	A implementação segue a maioria das especificações, com algumas pequenas variações ou omissões.	A implementação segue parcialmente as especificações, mas omite ou erra em pontos importantes.	A implementação falha em aspectos importantes das especificações, comprometendo a funcionalidade.	A implementação não segue as especificações de forma significativa, prejudicando a solução.
<b>Clareza de codificação</b>	O código é bem estruturado, com boa nomenclatura de variáveis, funções e comentários que facilitam o entendimento.	O código é estruturado de forma geral, mas poderia ser mais claro com alguns ajustes.	O código é razoável, mas possui áreas que poderiam ser mais legíveis ou organizadas.	O código é difícil de entender devido a falta de clareza ou nomeações inadequadas em algumas partes.	O código é confuso, mal estruturado e difícil de entender sem uma análise profunda.
<b>Critérios de escolha</b>	As escolhas de técnicas e soluções são bem fundamentadas, apropriadas e justificadas de forma clara.	As escolhas de técnicas são geralmente apropriadas, mas com algumas escolhas questionáveis.	Algumas escolhas não são as mais apropriadas, ou não há uma justificativa clara para as decisões.	As escolhas feitas não são as mais adequadas ou são pouco justificadas.	As escolhas de técnicas são inadequadas ou mal fundamentadas para o problema.
<b>2. Apresentação (01 ponto)</b>					
Critério	Atende perfeitamente (1,0)	Atende de forma satisfatória (0,8)	Atende parcialmente (0,6)	Atende de forma insatisfatória (0,3)	Não atende (0)
<b>Apresentação do trabalho</b>	A apresentação é clara, bem estruturada, com explicações detalhadas sobre a implementação e desafios enfrentados.	A apresentação está boa, mas poderia ser mais detalhada ou clara em alguns pontos.	A apresentação tem pontos importantes faltando ou não está completamente clara.	A apresentação é difícil de seguir e carece de explicações suficientes sobre a implementação.	A apresentação é confusa, sem uma estrutura clara ou explicações suficientes.