



**Pontifícia Universidade Católica de Minas Gerais**  
**Instituto de Ciências Exatas e Informática**  
**Algoritmos e Estruturas de Dados III - Prof. Felipe Soares**

**Trabalho Prático I**  
**5 pontos**

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

**Penalidade por atraso:** a cada dia corrido de atraso, a nota será penalizada em 1 ponto.

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

**Descrição:**

Neste trabalho, você deverá implementar um sistema responsável por realizar operações de CRUD (create, read, update e delete) em um arquivo sequencial. Para a realização dessas operações, você deve escolher entre dois contextos diferentes: clubes de futebol ou contas bancárias.

**Orientações:**

- O sistema deve ser implementado em C, C++, C# ou 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).
- Todo o código deve ser comentado de modo a se compreender a lógica utilizada. A não observância desse critério implica na redução da nota final em 50%.
- A organização do arquivo de dados deve ser sequencial (módulo 4 no Canvas). A estrutura do arquivo 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.



# Pontifícia Universidade Católica de Minas Gerais

## Instituto de Ciências Exatas e Informática

### Algoritmos e Estruturas de Dados III - Prof. Felipe Soares

- Os objetos utilizados devem possuir os seguintes atributos (cabe a você realizar as escolhas dos melhores tipos para cada atributo):
  - Caso a escolha seja clubes de futebol:
    - *idClube* (deve ser incremental à medida que novos registros forem adicionados)
    - *nome*
    - *cnpj*
    - *cidade*
    - *partidasJogadas*
    - *pontos*
  - Caso a escolha seja contas bancárias:
    - *idConta* (deve ser incremental à medida que novos registros forem adicionados)
    - *nomePessoa*
    - *cpf*
    - *cidade*
    - *transferenciasRealizadas*
    - *saldoConta*
- O sistema deverá oferecer uma tela inicial (com uso pelo terminal) com um menu com as seguintes opções:
  - Caso a escolha seja clubes de futebol:
    - Criar um clube -> essa escolha deve, a partir da leitura dos dados do clube pelo terminal (*nome*, *cnpj*, *estado*), criar um novo clube no arquivo (*partidasJogadas* e *pontos* devem ser iniciados com 0).
    - Realizar uma partida -> essa escolha deve atualizar dados em dois clubes no arquivo.
      - Para isso, é necessário permitir ao usuário cadastrar uma partida, ou seja, informar o nome de dois times e quantos gols fizeram. Cabe ao método criado definir o ganhador ou se foi empate. Caso tenha vencedor, deve-se somar 3 pontos para esse. Caso seja empate, deve-se somar 1 ponto para cada clube. Então, o programa deve atualizar o campo *pontos* e o campo *partidasJogadas* (em +1) dos dois clubes.
  - Caso a escolha seja contas bancárias:
    - Criar uma conta bancária -> essa escolha deve, a partir da leitura dos dados da conta bancária pelo terminal (*nomePessoa*, *cpf*, *estado*), criar uma nova conta no arquivo (*saldoConta* e *transferenciasRealizadas* devem ser iniciados com 0).
    - realizar uma transferência -> essa escolha deve atualizar dados em duas contas no arquivo.
      - Para isso, é necessário permitir ao usuário cadastrar uma operação de transferência, ou seja, informar duas contas (uma para debitar e outra para creditar o valor) e o valor da transferência. Assim, a conta para debitar deve ter uma redução em *saldoConta* do valor da transferência, enquanto que a conta para *creditar* deve receber um acréscimo. Então, o



# Pontifícia Universidade Católica de Minas Gerais

## Instituto de Ciências Exatas e Informática

### Algoritmos e Estruturas de Dados III - Prof. Felipe Soares

programa deve atualizar o campo *saldoConta* e o campo *transferenciasRealizadas* (em +1) das duas contas.

- Para ambas as escolhas:
  - Ler um registro (id) -> esse método deve receber um id como parâmetro, percorrer o arquivo e retornar os dados do id informado.
  - 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.

### O que deve ser entregue:

Deve ser entregue um relatório (usando LaTeX - utilizar “SBC Conferences Template”) como resultado da realização deste trabalho prático (máximo 10 páginas). Anexo a esse relatório deve ser entregue o projeto do sistema desenvolvido (formato .zip). Esse relatório deve conter a seguinte estrutura:

- Título
- Resumo
- Introdução
- Desenvolvimento
- Testes e Resultados
- Conclusão

### Critérios para avaliação

- Implementação do sistema (3,5 pontos)
  - Correção e robustez dos programas
  - Conformidade às especificações
  - Clareza de codificação
  - Critérios de escolha
- Documentação (1,5 pontos)