

## Introdução

O projeto propõe o desenvolvimento de um sistema de cadastro robusto, proporcionando funcionalidades essenciais como busca e alteração de dados, com a peculiaridade de manipulação de arquivos binários e .csv. Este projeto foi conduzido por um integrante: Lucas Torquato. O sistema de cadastro é concebido para lidar com informações sobre astros de Hollywood, incorporando uma estrutura de dados que abrange, no mínimo, cinco campos de diferentes tipos, incluindo representações textuais e numéricas: identificador, nome, principal filme feito e quantidade de óscares. A implementação prioriza a eficiência por meio de práticas como alocação dinâmica e redimensionamento de vetores. A manipulação de arquivos binários e .csv é a espinha dorsal do armazenamento persistente.

O funcionamento do código se dá por um sistema simples de gerenciamento de dados relacionados a nomes de atores famosos de Hollywood, utilizando uma struct chamada “Dados” para armazenar informações como: id, nome do ator, data de nascimento, título do filme e a quantidade de prêmios Óscar recebidos. O programa oferece um menu interativo para realizar diversas operações, como importar dados de um arquivo CSV, inserir novos registros, remover registros existentes, fazer buscas por nome de ator(a) ou por id, ordenar por ID e por nascimento, imprimir a lista com ou sem limite e também sair do programa, além de outras manipulações.

O fluxo principal do programa está na função `main`, onde o usuário é guiado por um menu que permite escolher entre diferentes opções. A estrutura do menu destaca a flexibilidade do programa, atendendo a diversas necessidades de gerenciamento de dados, desde consultas simples até operações mais avançadas. Essa abordagem torna o sistema versátil e prático para aqueles que buscam uma solução completa de cadastro e manipulação de informações sobre atores de Hollywood.

## Descrição das funções e aplicações fora da `main`

### Struct `Dados`

Esta estrutura armazena os dados referentes a um ator, incluindo:

- Identificador (id)
- Nome do ator (ator)
- Ano de nascimento (nascimento)
- Nome do filme em destaque (filme)
- Quantidade de prêmios Óscar conquistados (oscar)

### Funções principais

- `imprimir`: Recebe um objeto do tipo `Dados` e imprime os valores de seus campos formatados.
- `imprimeVetor`: Recebe um array de objetos `Dados` e seu tamanho, iterando sobre o array e chamando a função `imprimir` para cada elemento.
- `ler`: Lê os dados de um arquivo CSV e os armazena nos objetos da estrutura `Dados`.

- **inserirRegistro**: Permite a inserção de um novo registro no sistema.
- **removerRegistro**: Remove um registro do sistema com base no ID.
- **buscarPorID e buscarPorNome**: Realizam a busca de registros com base no identificador (ID) ou no nome do ator.
- **ordenaPorID e ordenaPorNascimento**: Ordenam os registros com base no ID e no ano de nascimento utilizando o algoritmo Shell/Insertion Sort.

## Descrição do menu

O sistema apresenta um menu interativo que permite ao usuário realizar diversas operações:

1. **Imprimir Lista**: Exibe todos os registros presentes no sistema.
2. **Imprimir com Limite**: Permite imprimir uma parte específica da lista.
3. **Inserir Novo Registro**: Adiciona um novo registro ao sistema.
4. **Remover Registro**: Remove um registro com base no ID informado.
5. **Buscar por ID**: Realiza uma busca sequencial no sistema com base no ID fornecido.
6. **Buscar por Nome**: Efetua uma busca sequencial no sistema com base no nome do ator.
7. **Transformar em Binário**: Converte os dados para um formato binário e os escreve em um arquivo.
8. **Ordenar por ID**: Utiliza o algoritmo Shell/Insertion Sort para ordenar os registros pelo identificador.
9. **Ordenar por Nascimento**: Ordena os registros pelo ano de nascimento.
10. **Ler Arquivo Binário**: Transfere os dados do arquivo binário para o vetor de registros.
11. **Sair**: Encerra a execução do programa e libera a memória alocada.

## Dados do Arquivo CSV

A ordem dos dados armazenados no arquivo CSV segue a estrutura definida na struct **Dados**. Cada linha no arquivo representa um registro e contém os seguintes campos, separados por vírgula:

- **ID** (inteiro)
- **Nome do ator** (texto)
- **Ano de nascimento** (inteiro)
- **Filme** (texto)
- **Quantidade de Óscares** (inteiro)

Cada registro é representado por uma linha no arquivo CSV, e os campos são separados por vírgulas, facilitando a leitura e escrita dos dados.

## Erros e Acertos

Ao longo do desenvolvimento do projeto, algumas dificuldades foram encontradas, especialmente na manipulação de arquivos binários.

## Conclusão

O sistema de cadastro desenvolvido oferece uma experiência prática e abrangente no desenvolvimento de aplicações de cadastro. A manipulação de

arquivos é um dos aspectos centrais, permitindo leitura e gravação em formatos CSV e binário. O gerenciamento dinâmico de memória destaca boas práticas de alocação e liberação de recursos, enquanto a implementação de algoritmos de busca e ordenação enfatiza a eficiência computacional.

Além disso, o sistema serve como um excelente exercício de aprendizado, aplicando conceitos fundamentais de C++ como struct, manipulação de arquivos, ordenação, busca, redimensionamento de vetores, alocação dinâmica e ponteiros. Assim, o projeto não apenas entrega um sistema de cadastro funcional, mas também constitui uma valiosa oportunidade de consolidação de conhecimentos em programação.