

TRABALHO FINAL – Cadastro de Filmes

REGRAS DO TRABALHO:

- 1) Equipe: até 3 participantes;
- 2) A equipe deverá implementar um programa que faça uso de tipos estruturados e arquivos, conforme enunciado a seguir;
- 3) O programa será avaliado por sua corretude, organização e aplicabilidade dos conceitos e técnicas de programação de forma correta (detalhes dos critérios ao final do enunciado);
- 4) O trabalho deve ser apresentado pela equipe **até 28/07/2022**

ESTRUTURAS DE DADOS:

O programa deve permitir o cadastramento de filmes conforme o seguinte modelo de dados:

- **Diretor:**
 - nome do diretor (cadeia de caracteres);
 - nacionalidade do diretor (cadeia de caracteres);
- **Data:**
 - dia, mês e ano (3 valores inteiros);
- **Filme:**
 - nome do filme (cadeia de caracteres) – **CAMPO-CHAVE**;
 - duração (em minutos, valor inteiro);
 - estilo (cadeia de caracteres);
 - dados do **diretor** (conforme estrutura especificada acima);
 - **data** do cadastramento (conforme estrutura especificada acima);

DESCRIÇÃO DO TRABALHO:

- A partir do modelo de dados apresentado acima, é preciso construir um programa que permita o gerenciamento básico deste **cadastro simples com a utilização de arquivos de texto**. O programa, portanto, carrega os dados de um arquivo de texto, e mostra um menu com as seguintes opções:
 - 1) **Inserir** novos registros (um de cada vez);
 - 2) **Remover** registros existentes (um de cada vez);
 - 3) Exibir todo o conteúdo do cadastro (**listagem completa**, um registro em cada linha);
 - 4) **Consultar** registros pelo campo "chave";
- O menu deverá ser implementado na função principal e as opções devem estar em funções separadas.
- O **campo chave** é o campo da estrutura principal que identifica o registro como único, portanto, não pode ter valores repetidos.
- A opção de **inserir** deve permitir cadastrar UM novo registro de cada vez, após isso o programa volta a apresentar o menu. A inserção não deve permitir cadastrar o campo "chave" repetido, isto é, se o registro já tiver sido cadastrado, deve mostrar uma mensagem de aviso e solicitar os dados novamente.

- A opção de **remover** deve perguntar qual registro deve ser apagado, através do campo “chave” que identifica o registro. Caso o usuário digite um registro inexistente, deve mostrar uma mensagem de aviso e pedir a informação novamente. Sugestões para implementar a remoção:
 - i. “Movimente” os dados do vetor, de modo a sobrescrever o registro a ser apagado, e realoque o vetor;
 - ii. Uma alternativa mais simples consiste em adicionar um campo *booleano* no registro de dados, indicando se o mesmo foi removido. Neste caso, o registro removido não é, de fato, removido da memória, mas sim marcado como tal. Portanto, todo o processamento (inserção, consulta, listagem, salvamento no arquivo, ...) deve ignorar os registros marcados como removidos.
- A opção de **listar** (exibir todos os registros) deve mostrar um relatório (**em tela** ou em um **arquivo texto**, conforme escolha do usuário) com um cabeçalho e com um registro em cada linha (se possível). Repare que o arquivo de texto gerado nesta funcionalidade conterá um relatório. Portanto, não confunda com o arquivo de texto que contém o cadastro dos dados. Veja modelo do relatório a seguir:

Listagem de Filmes Cadastrados

Filme	Diretor	Nacionalidade	Cadastramento
-----	-----	-----	-----
Tubarao	Spielberg	Americana	11/10/2020
Tropa de Elite	Jose Padilha	Brasileira	05/05/2021
A Origem	Chris Nolan	Britanica	15/11/2021

- A opção de **consultar** um registro deve perguntar qual registro o usuário deseja visualizar através do campo “chave”. Caso o usuário digite um registro inexistente, deve mostrar uma mensagem de aviso e pedir a informação novamente.
- O menu deve apresentar uma opção para **sair** do programa.
- Use e abuse de funções por valor ou por referência, mas em nenhuma hipótese faça uso de variáveis globais. Faça uso de funções da biblioteca padrão do C, conforme a necessidade.
- Quando iniciado, o programa deve carregar todo o arquivo em um vetor de estruturas. Caso o arquivo não exista, o programa gera um vetor “vazio” (ponteiro *NULL*). Ao final da execução do programa, o vetor é gravado no arquivo, substituindo completamente o conteúdo original (se o arquivo existir).
- O formato do arquivo de cadastro pode ser o mais simples possível: a primeira linha contém a quantidade de registros (filmes) cadastrados; cada linha subsequente contém o valor de cada campo (conforme exemplo feito em aula);
- O programa deve lidar com a realocação do vetor, pois não há limite para a quantidade de registros.
- Não se esqueça de desalocar todas as estruturas ao final da execução do programa.

Critérios de avaliação:

Item avaliado	Detalhes	Pontuação
Menu	Implementação correta do menu na função principal, fazendo com que sempre seja reapresentado ao usuário ao final de cada operação; as opções devem ser implementadas em funções separadas.	1,0
Inserir	Implementação correta função inserir, garantindo que os novos dados digitados sejam inseridos na estrutura e gravados adequadamente no arquivo; também deve garantir que não seja inserido um registro com um campo-chave que já existe (mesmo filme não deve ser inserido duas vezes), indicando este erro para o usuário.	1,5
Listar todos	Implementação correta da função listar todos, onde somente os dados indicados no enunciado devem aparecer, com um registro em cada linha, mostrando todos os filmes já cadastrados; se o usuário escolher esta opção mas não houver dados no arquivo, deve aparecer uma mensagem indicando esta situação.	1,5
Consultar	Implementação correta da função consultar, onde o usuário deve indicar o nome de um único filme; se o filme estiver cadastrado, todos os dados dele devem aparecer na tela; caso contrário, deve mostrar uma mensagem indicando que o filme não existe no arquivo.	1,5
Remover	Implementação correta da função remover, onde o usuário deve indicar o nome de um único filme; se o filme estiver cadastrado, os dados dele devem ser excluídos (não devem mais aparecer para o usuário); caso o filme indicado não esteja cadastrado, mostrar uma mensagem indicando esta situação.	1,5
Extras	Uso correto do modelo de dados (estruturas), aparência do programa para o usuário, tratamento dos dados do arquivo, uso de alocação de memória.	1,5
Defesa	Cada estudante deverá responder a perguntas feitas pela professora durante a apresentação, indicando o conhecimento e entendimento do código apresentado.	1,5