

Trabalho Final

Professor: Josemar Alves Caetano

Data de entrega: 13 de Fevereiro de 2022

Instruções: leia com atenção **todas** as orientações abaixo antes de começar o trabalho. **Importante:** Este trabalho deve ser feito em grupos de **5 a 7 pessoas** e vale 30 pontos na disciplina teórica e 30 na prática.

Carlos é um grande entusiasta de filmes. Sua paixão pela 7ª arte vai além de somente assisti-los. Ele gosta de escrever *reviews*, colecionar posters e sempre que possível discutir com seus amigos cinéfilos à respeito dos filmes que assiste. Além disso, Carlos pretende levar a discussão de filmes para além dos limites da sua roda de amizade. Ele deseja construir um fórum de discussão online. Para isso, ele contratou você e sua equipe para implementar um programa que possibilite o gerenciamento de uma base de filmes.

Seu programa deve implementar:

- Uma struct chamada **Filme**
 - A **struct Filme** é um registro que armazena as informações de um filme da base. Ela possui os seguintes atributos:
 - * **identificador** - é um número que deve ser único para cada filme e deve ser atribuído automaticamente.
 - * **nome** - máximo 50 caracteres.
 - * **genero** - máximo 30 caracteres.
 - * **anoLancamento** - deve ser um número entre 1900 e 2021.
 - * **nomeDiretor** - máximo 30 caracteres.
 - Você pode acrescentar outros atributos se achar necessário.
- Sete funções cujos nomes são:
 - **main**: esta função deve apresentar um menu com as opções para acessar as demais funcionalidades do sistema e direcionar o fluxo de execução.
 - **adicionaFilme**: esta função adiciona um novo filme ao vetor, após receber os dados do usuário.
 - **editaFilme**: altera os dados de um filme.
 - **removeFilme**: remove um filme do vetor.
 - **imprimeFilmes**: imprime o vetor de filmes cadastrados. **Desafio (1 ponto extra):** Implemente a função **imprimeFilmes** de forma que ela possa imprimir todos os filmes ou apenas um filme escolhido pelo usuário.
 - **gravaFilmes** que grava em um arquivo o vetor de filmes cadastrados.
 - **leFilmes** que lê um vetor de filmes de um arquivo e cria o vetor de filmes cadastrados no programa.

1 Avaliação

- Para a disciplina **prática**: o código fonte em **C** com as soluções do trabalho.

Seu código deve estar bem organizado, indentado e conter comentários relevantes.

Faça esta tarefa no **Dev-C++** ou em algum software similar e submeta no SIGAA (tarefa da disciplina de laboratórios):
- Para a disciplina **teórica**: um relatório em formato **doc** ou **pdf** explicando cada funcionalidade do sistema e como foram aplicados os conceitos vistos na disciplina.

Elabore dois fluxogramas: (i) da função `editaFilme` e (ii) da função `removeFilme`.

O relatório deve conter no máximo 4 páginas (exceto imagens).

Desafio (3 pontos extras): Elabore um fluxograma geral do sistema.

Submeta o relatório com os fluxogramas no SIGAA (tarefa da disciplina teórica).

- **Importante:** Apenas um aluno precisa realizar o envio. Porém, os nomes de **todos** os integrantes devem estar comentados na primeira linha do programa e no relatório. Não precisa ser o mesmo integrante a realizar as duas submissões.
- Cópias de trabalho são inaceitáveis, podem ser detectadas automaticamente e receberão nota zero.
- Códigos fonte em PDF e submissões em atraso serão desconsideradas.
- Qualquer indício de cópia resultará no **anulação** de todo o trabalho. É importante ressaltar que todos os códigos serão inseridos em um detector de plágio que identifica os arquivos duplicados.

2 Critérios de avaliação

Os trabalhos serão avaliados de acordo com os critérios:

- Implementação correta [0 a 15 pontos].
- Tratamento de erros, eficiência e elegância [0 a 5 pontos].
- Organização do código: clareza e documentação [0 a 5 pontos].
- Interface com usuário [0 a 5 pontos]:
 - Usabilidade, criatividade e estética.
- Desafios terão nota 0, 25%, 50%, 75% ou 100% dos pontos extras de acordo com o funcionamento e adequação ao restante do código.
- **Cópias terão nota zero!**
- Trabalhos que não compilarem terão notas entre 0 e 5.
- Caso necessário, haverá uma entrevista com os integrantes do grupo.

3 Desafio (disciplina prática)

(3 pontos extras) Implemente a lógica de armazenamento das revisões, notas e discussões de cada filme. Para isso, seu programa deve **adicionalmente** implementar:

- Uma struct chamada **Revisao**
- Duas funções cujos nomes são:
 - **insereRevisao**: esta função adiciona uma nova revisão de um determinado filme da base, após receber os dados do usuário. A função deve gravar os dados diretamente em arquivo.
 - **listaRevisoes**: imprime todas as revisões de um determinado filme cadastrado. **Mais um desafio (1 ponto extra):** Implemente a função **listaRevisoes** de forma que ela possa imprimir todas as revisões de um filme ou apenas revisões feitas por um usuário específico em um filme. Os dados devem ser lidos de um arquivo e podem ser armazenados temporariamente em um vetor (variável local).

struct Revisao

A **struct Revisao** é um registro que armazena as informações de uma revisão de um filme da base. Ela possui os seguintes atributos:

- **identificador** - é um número que deve ser único para cada revisão e deve ser atribuído automaticamente.

- **identificadorFilme** - número de identificação único de cada filme.
- **nomeUsuario** - máximo 50 caracteres.
- **texto** - máximo 500 caracteres.
- **data** - deve ser do tipo de uma struct que tem como atributos: dia, mes e ano.