

## **Projeto Maromo**

Gerenciador de Catálogo de Filmes em C

Membros:

Edvan dos Santos Noronha

Felipe Lopes do Santos

Kaio Estevam Pereira Dias

Lucas de Campos Gonçalves

Paulo Renato Lilli

Ruan Modena dos Santos

Vinicius Barboza Leme

Curso/Disciplina: Linguagem de Programação

Data: 04 de Julho de 2025

## Sumário

Visão Geral do Projeto.....	3
Funcionalidades.....	3
Estrutura do Projeto .....	3
Estrutura de Dados.....	4
Compilação e Execução.....	5
Descrição das Funções (API) .....	5
Formato do Arquivo de Dados ( <code>filmes.csv</code> ) .....	7

## Visão Geral do Projeto

O "Catálogo de Filmes" é uma aplicação de console que permite ao usuário gerenciar uma coleção de filmes. O programa possibilita adicionar, listar, buscar e atualizar informações sobre filmes, que são armazenadas em um arquivo CSV. O objetivo principal é fornecer uma interface simples e direta para a manipulação de um catálogo de filmes pessoal, com persistência de dados.

## Funcionalidades

O sistema oferece as seguintes funcionalidades através de um menu interativo:

**Adicionar filme:** Permite ao usuário inserir um novo filme no catálogo, fornecendo informações como ID, título, diretor, gênero, ano de lançamento e avaliação. O sistema valida os dados de entrada, como o ano e a nota da avaliação.

**Buscar por diretor:** Realiza uma busca por todos os filmes de um diretor específico, exibindo uma lista dos filmes encontrados.

**Filtrar por gênero:** Exibe todos os filmes que correspondem a um gênero informado pelo usuário.

**Atualizar avaliação:** Permite modificar a nota de avaliação de um filme existente, identificado pelo seu ID.

**Listar filmes:** Exibe todos os filmes do catálogo, ordenados por ano de lançamento e, secundariamente, por título.

**Buscar por nome do filme:** Procura por um filme específico a partir do seu título.

**Sair:** Encerra o programa e salva todas as alterações realizadas no arquivo de dados.

## Estrutura do Projeto

O projeto é composto pelos seguintes arquivos:

**main.c:** Contém a função `main`, que inicializa o programa, e a função `menu`, que exibe as opções ao usuário e gerencia a interação, direcionando para as funções apropriadas com base na escolha do usuário.

**filmes.h:** Arquivo de cabeçalho que define a estrutura de dados `Filme` e os protótipos de todas as funções implementadas em `filmes.c`. Também define constantes importantes como o número máximo de filmes (`MAX_FILMES`) e o tamanho máximo de strings (`MAX_STR`).

**filmes.c:** É o coração do projeto. Contém a implementação de todas as funcionalidades, como a inicialização do catálogo a partir do arquivo CSV, o salvamento dos dados, a adição, busca e listagem de filmes, além de funções auxiliares para leitura e validação de dados.

**filmes.csv:** Arquivo de dados no formato CSV (Comma-Separated Values) que armazena as informações dos filmes. Cada linha representa um filme, e os campos são separados por vírgulas.

**CMakeLists.txt:** Arquivo de configuração para o sistema de compilação CMake. Ele define o nome do projeto, o padrão da linguagem C a ser utilizado e especifica quais arquivos-fonte (`main.c` e `filmes.c`) devem ser compilados para gerar o executável final chamado `Projeto_Maromo`.

## Estrutura de Dados

A principal estrutura de dados do projeto é a `struct Filme`, definida no arquivo `filmes.h`. Ela armazena todas as informações pertinentes a um filme:

C

```
typedef struct {
    int id;
    char titulo[MAX_STR];
    char diretor[MAX_STR];
    int ano;
    char genero[MAX_STR];
}
```

```
float avaliacao;  
} Filme;
```

`id`: Um número inteiro único que identifica o filme.

`titulo`: O título do filme.

`diretor`: O nome do diretor do filme.

`ano`: O ano de lançamento do filme.

`genero`: O gênero cinematográfico do filme.

`avaliacao`: A nota de avaliação do filme, em um formato de ponto flutuante.

## Compilação e Execução

Para compilar e executar o projeto, é necessário ter o CMake e um compilador C (como o GCC) instalados. Siga os passos abaixo:

Crie um diretório para a compilação (por exemplo, `build`).

Navegue até este diretório.

Execute o CMake para gerar os arquivos de compilação: `cmake . .`

Compile o projeto: `make`

O executável `Projeto_Maromo` será criado. Para executá-lo, utilize o comando: `./Projeto_Maromo`.

O arquivo `CMakeLists.txt` especifica que o projeto deve ser compilado usando o padrão C11 e que os arquivos `main.c` e `filmes.c` são necessários para criar o executável.

## Descrição das Funções (API)

A seguir, uma descrição detalhada das principais funções implementadas em `filmes.c`:

**void inicializa():** Carrega os dados dos filmes do arquivo `filmes.csv` para a memória. Abre o arquivo, lê cada linha, e popula a lista de filmes.

**void salvarDados():** Salva o estado atual do catálogo de filmes de volta para o arquivo `filmes.csv`, sobrescrevendo o conteúdo anterior. É chamada após operações de adição ou atualização.

**void adicionarFilme():** Solicita ao usuário todas as informações de um novo filme, realiza validações (como a unicidade do ID e a faixa de valores para ano e avaliação), aloca memória para o novo filme e o adiciona ao catálogo.

**void buscarPorDiretor():** Pede ao usuário o nome de um diretor e percorre a lista de filmes, exibindo todos aqueles cujo diretor corresponde à busca (ignorando a diferença entre maiúsculas e minúsculas).

**void filtrarPorGenero():** Solicita um gênero e exibe todos os filmes que pertencem a esse gênero. A comparação é sensível a maiúsculas e minúsculas.

**void atualizarAvaliacao():** Pede o ID de um filme e, se encontrado, permite que o usuário insira uma nova nota de avaliação.

**void listarFilmes():** Ordena a lista de filmes usando a função `qsort` com base no ano e no título. Em seguida, exibe a lista completa e ordenada.

**void buscarPorNome():** Pede ao usuário o título de um filme e exibe as informações do filme correspondente, caso seja encontrado. A busca ignora maiúsculas e minúsculas.

### **Funções Auxiliares:**

`limparBuffer():` Limpa o buffer de entrada do teclado.

`lerString():` Lê uma string do usuário, removendo o caractere de nova linha.

`lerInteiro()`: Lê um número inteiro do usuário, com validação para garantir que apenas números sejam inseridos.

`idExiste()`: Verifica se um determinado ID já está em uso no catálogo.

`comparar()`: Função de comparação usada pelo `qsort` para ordenar os filmes.

### Formato do Arquivo de Dados (`filmes.csv`)

O arquivo `filmes.csv` é a base de dados do sistema e segue um formato específico:

A primeira linha é o cabeçalho, descrevendo as colunas:  
`ID,Titulo,Diretor,Ano,Genero,Avaliação`.

Cada linha subsequente representa um filme.

Os campos de cada filme são separados por vírgulas, na seguinte ordem:  
`ID, Título, Diretor, Ano, Gênero, Avaliação`.