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

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.

Funções

inicializa()

Objetivo: Carrega os dados do arquivo CSV (filmes.csv) para a memória ao iniciar o programa.

Como funciona:

Abre o arquivo CSV em modo leitura.

Lê cada linha do arquivo e preenche um vetor de structs Filme.

Atualiza a variável de controle com a quantidade de filmes carregados.

Tratamento de erros: Se o arquivo não existir, cria um novo arquivo vazio.

---

salvarDados()

Objetivo: Grava no arquivo CSV os dados atualizados da memória.

Como funciona:

Abre o arquivo CSV em modo escrita.

Percorre o vetor de filmes.

Escreve cada filme em uma linha do arquivo CSV.

Tratamento de erros: Exibe mensagem se ocorrer falha ao abrir ou escrever no arquivo.

---

adicionarFilme()

Objetivo: Adiciona um novo filme ao catálogo.

Como funciona:

Solicita dados do filme (ID, título, diretor, ano, gênero, avaliação) ao usuário.

Preenche um novo elemento no vetor de filmes.

Incrementa a quantidade total de filmes.

Chama `salvarDados()` para persistir a inclusão.

Validações: Confirma se não há excesso de filmes (verificação do limite `MAX_FILMES`).

---

listarFilmes()

Objetivo: Exibe todos os filmes cadastrados, ordenados por ano e título.

Como funciona:

Utiliza a função `qsort` para ordenar o vetor de filmes.

Imprime na tela todos os dados dos filmes ordenados.

Critério de ordenação: Ano crescente; títulos em ordem alfabética se anos forem iguais.

---

buscarPorDiretor()

Objetivo: Exibe todos os filmes dirigidos por um determinado diretor.

Como funciona:

Solicita o nome do diretor ao usuário.

Percorre o vetor de filmes.

Compara o campo `diretor` usando `strcasecmp` para ignorar maiúsculas/minúsculas.

Exibe informações dos filmes encontrados.

---

buscarPorNome()

Objetivo: Busca um filme específico pelo título.

Como funciona:

Solicita o título do filme ao usuário.

Percorre o vetor de filmes.

Compara o campo `titulo` utilizando `strcasecmp`.

Exibe detalhes do filme encontrado, ou mensagem caso não exista.

---

filtrarPorGenero()

Objetivo: Exibe apenas filmes de um gênero específico.

Como funciona:

Solicita ao usuário o gênero desejado.

Percorre o vetor de filmes.

Compara o campo genero usando strcasecmp.

Exibe a lista filtrada.

---

atualizarAvaliacao()

Objetivo: Atualiza a nota (avaliação) de um filme já cadastrado.

Como funciona:

Solicita o ID do filme ao usuário.

Verifica se o ID existe no vetor de filmes.

Solicita nova avaliação e atualiza o campo avaliacao.

Chama salvarDados() para gravar a alteração.

Validações: Confirma se a nova avaliação está dentro do intervalo permitido (0 a 10).

---

lerInteiro()

Objetivo: Ler valores inteiros da entrada padrão de forma segura.

Como funciona:

Lê uma string digitada pelo usuário.

Verifica se todos os caracteres são dígitos ou sinal negativo.

Converte a string para inteiro apenas se for válida.

Tratamento de erros: Repete a solicitação ao usuário em caso de entrada inválida.

---

menuPrincipal()

Objetivo: Exibe o menu e chama as funções correspondentes conforme a escolha do usuário.

Como funciona:

Mostra as opções disponíveis no console.

Lê a opção escolhida.

Executa a função referente à opção até que o usuário escolha sair.

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.