UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO E
ENGENHARIA DE COMPUTAÇÃO

ALUNO(S) AUTOR(ES)
Luís Filipe Santos de Moura
Pedro Schuck de Azevedo

Análise e Desenvolvimento de um Sistema de Catalogação Cinematográfica

Relatório apresentado como requisito parcial para a obtenção de conceito na Disciplina de Classificação e Pesquisa de Dados.

Orientador: Prof. Dr. Leandro Krug Wives

Porto Alegre,
2025

SUMÁRIO

1 INTRODUÇÃO

Um catálogo é uma estrutura que permite ao usuário visualizar, pesquisar, filtrar, inserir e ordenar dados de um determinado tipo, sendo útil justamente por armazenar em um mesmo local uma grande quantidade de informações sobre múltiplos elementos. No meio digital, catálogos utilizam de bancos de dados disponibilizados por diferentes entidades como base para dispor estes dados de maneira organizada e ordenada, além de permitir uma série de operações para facilitar a navegação do usuário. Todavia, devido ao número expressivo de dados e o espaço massivo que eles podem ocupar na memória, torna-se necessário tomar várias medidas para deixá-los mais enxutos, através de codificação, por exemplo, e para otimizar ao máximo os algoritmos que lidam com eles, como trabalhar com blocos de elementos ao invés de tratá-los singularmente.

Um filme ou obra cinematográfica é um meio de entretenimento audiovisual, que consiste em uma sequência de imagens projetadas rapidamente para dar a ilusão de movimento, tendo como intuito contar uma história, expressar emoções, retratar a realidade e entre outras finalidades. Desde a criação do primeiro filme no ano de 1888, os longa-metragens evoluíram significativamente ao longo das décadas, passando a possuir som, cores, efeitos especiais, computação gráfica e várias outras técnicas que aprimoraram e auxiliram eles na sua popularização, tornando-os parte da cultura atual e até mesmo uma forma internacionalmente reconhecida de arte. Levando em consideração a relevância histórica e social dos filmes, criou-se a proposta de desenvolver um sistema que catalogue a maior quantidade possível de películas, permitindo todas as operações padrões, bem como todas aquelas adequadas e necessárias ao contexto deste projeto, não só para ajudar o grande público a encontrar novos títulos e compará-los entre si, mas também para manter viva a paixão pelas obras cinematográficas.

2 DESENVOLVIMENTO

Devido a certa complexidade envolvida na tarefa de construir um catálogo cinematográfico, bem como para fins de organização, ela foi dividida em etapas, que representam de maneira cronológica os passos necessários para sua criação. De maneira geral, estas etapas podem ser descritas como: extração de dados, preparação dos dados, implementação da árvore B e arquivos invertidos como arquivos de índice, codificação da busca, filtragem, ordenação, inserção, exibição de estatísticas e interface para o usuário.

2.1 Extração de Dados

Os dados foram extraídos do site TMDb (The Movie Database), uma base de dados gratuita e de código aberto que armazena inúmeros registros de filmes e séries em diferentes idiomas. O acesso às informações necessárias foi adquirido fazendo-se um cadastro na plataforma do TMDb para obter uma chave API, que serve para autenticar e identificar a solicitação de um usuário a uma determinada aplicação, no caso em questão, para coletar os dados do TMDb.

Uma vez aceita a solicitação, o próprio TMDb disponibilizou um arquivo json compactado com todas as produções cinematográficas de sua base de dados, totalizando mais de um milhão destas. Neste arquivo, havia poucas informações sobre cada filme, sendo a mais relevante delas o seu identificador. Todavia, usando um programa em python, todos os identificadores foram extraídos, e por meio destes, foi possível utilizar uma ferramenta da API que faz o download das informações de uma película através de seu identificador. Desta forma, foi obtida uma quantidade mais expressiva de informações sobre cada obra, sendo que todas essas informações foram armazenadas em um arquivo no formato ndjson, onde cada linha do arquivo corresponde a um json com os dados de um título cinematográfico.

2.2 Preparação dos Dados

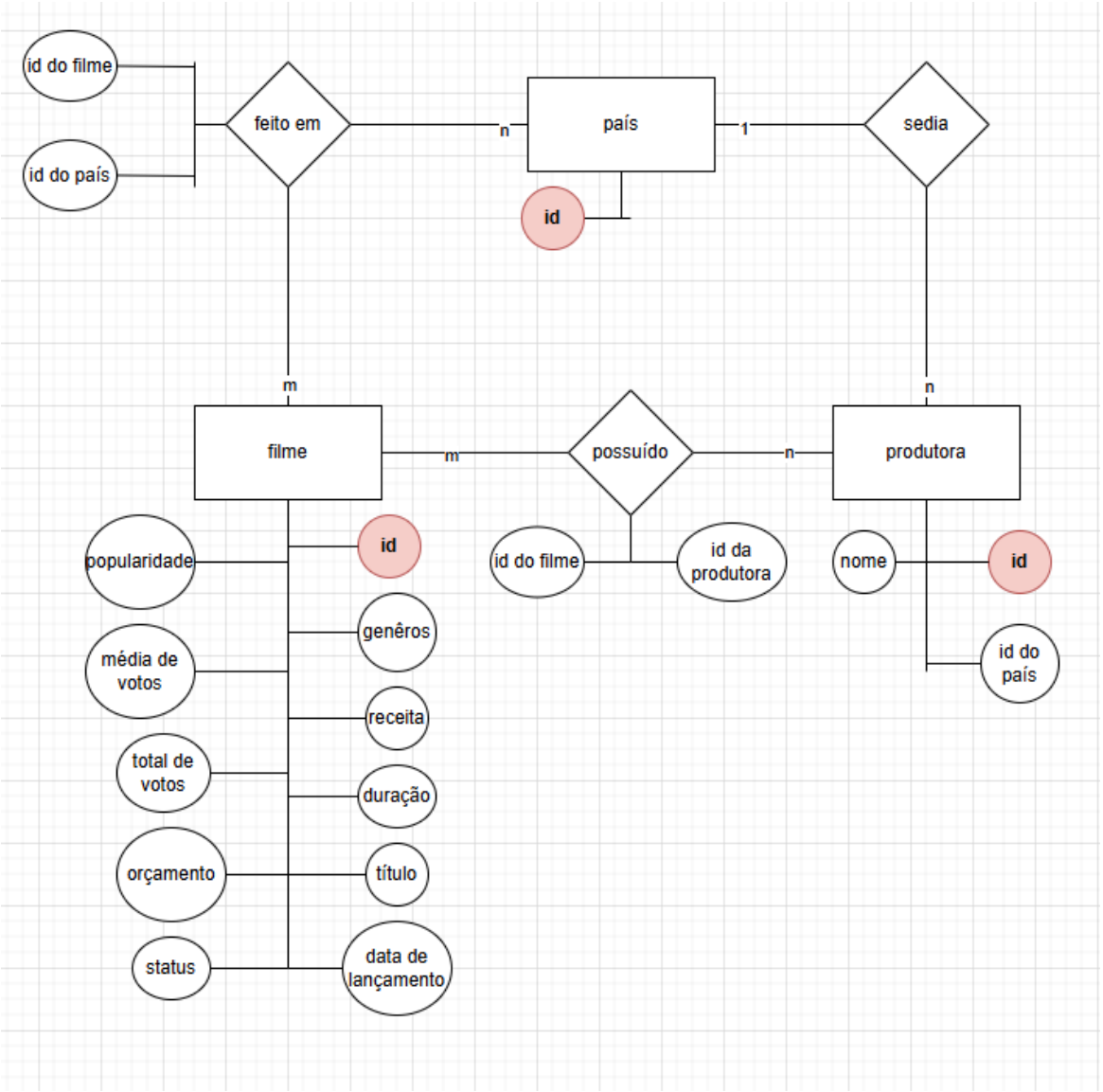
Uma vez que os dados foram obtidos, tornou-se necessário tratá-los, selecionando quais informações seriam mais úteis para a aplicação, que trabalharia

unicamente com arquivos binários próprios do tipo serial, isto é, cujos elementos não seguem nenhuma ordem específica, por fins de objetividade e simplicidade. Certos campos, como link para site, para o poster e descrição do filme foram cortados por serem considerados sem muita utilidade e demasiadamente grandes, ocupando muitos bytes de memória, respectivamente. Outros como data de lançamento e gênero foram codificados para serem armazenados em um único número inteiro. Em relação à data, os quatro dígitos mais significativos representam o ano, os dois menos significativos o dia e os dois restantes o mês de lançamento da película em questão. Para codificar os gêneros, utilizou-se um esquema de representação binário com 19 bits, no qual cada um retrata um tipo distinto, onde se o bit estiver em 1, significa que a obra possui aquele gênero, caso contrário, não, assim permitindo designar múltiplas categorias para o mesmo filme.

Foi feito uso da biblioteca ctypes em python para salvar as informações de cada filme de maneira semelhante a uma struct na linguagem de programação C. Com uma função que converte estas structs para uma sequência de bytes, as informações das obras foram preservadas em um arquivo binário. De maneira análoga, foi criada uma struct para conservar as informações de cada produtora, cujos elementos foram posteriormente escritos em um arquivo binário. Este mesmo programa implementado em python gerou arquivos que armazenavam as relações filme-produtora e filme-país, além de ter criado dois arquivos que guardam, respectivamente, os títulos de cada filme e os nomes de cada produtora.

Determinados aspectos julgados importantes, mas que apresentavam estrutura e campos próprios independentes do filme em si foram separados do arquivo binário principal, porém mantendo-se uma forma de conectar ambos entre si. Este foi o caso das produtoras de um filme, onde fazendo uso do sistema de Entidade e Relacionamento (Figura 2.1), estabeleceu-se que a relação filme-produtora é n para n, isto é, vários filmes podem possuir muitas produtoras, o que incentivou a criação de um arquivo contendo uma tabela onde cada elemento é um par (filme, produtora), permitindo várias ocorrências de uma película, cada uma com uma produtora diferente, no caso desta apresentar múltiplas produtoras. Pela forma como esta tabela foi construída, todas as ocorrências de uma obra estão em sequência, e valendo-se deste fato, guarda-se apenas o offset em bytes da primeira ocorrência do título cinematográfico na tabela do arquivo da relação filme-produtora dentro da struct de cada filme.

Figura 2.1 – Diagrama de Entidade e Relacionamento para a estrutura filme



Fonte: Elaboração própria.

Em relação aos títulos dos filmes e aos nomes das produtoras, optou-se por criar arquivos separados para eles, tendo em vista a alta irregularidade no tamanho destas strings. Exemplificando, verificou-se que haviam, apesar de serem poucos, títulos e nomes de produtoras com mais de 200 caracteres, bem como alguns com apenas 20. Para evitar o desperdício e ainda manter a informação dos títulos completa, usou-se um sistema de blocos de 32 bytes para armazenar os títulos e nomes, no qual o último byte de cada bloco indicava a ocupação do próximo bloco subsequente para o mesmo nome. Além disso, cada struct do tipo filme e produtora

carregam consigo o offset do seu nome nos seus respectivos arquivos binários, para poder acessá-los.

2.3 Arquivos de Índice

Levando em consideração o fato dos dados principais dos filmes estarem em um arquivo binário serial, percebeu-se que as operações levariam um tempo considerável para serem realizadas. Uma busca pelo nome de uma película, por exemplo, teria que ser feita de modo linear, verificando-se todos os títulos até encontrar o desejado. Portanto, utilizaram-se dois tipo de arquivos de índice, cuja finalidade é justamente acelerar as manipulações sobre os dados ao realizar uma organização destes de acordo com alguma categoria (nome, gênero, data de lançamento e etc), mantendo em cada elemento apenas o valor correspondente a esta para cada filme e a posição em que este se encontra no arquivo original. Desta forma, é possível realizar uma busca mais eficiente, como através de árvores B, para localizar determinado filme ordenado em certa categoria, fazendo uso da posição no arquivo original para acessar os demais dados associados.

2.3.1 Árvores B

Trata-se de uma estrutura de dados muito empregada como arquivo de índice por facilitar o tratamento de grandes volumes de dados, especialmente quando contidos em memória secundária. Algumas de suas principais características que justificam tais aspectos são o fato de ela sempre manter-se balanceada, com todos nodos folhas no mesmo nível, e que cada um de seus nodos pode armazenar múltiplas chaves, diferentemente de outros tipos de árvores, que servem como separadoras de intervalo para seus filhos. Assim, um vértice pode armazenar um bloco inteiro da memória e uma busca está limitada, no máximo, a percorrer a altura de uma árvore, o que possui um custo assintótico de $O(\log_n)$.

Usou-se de árvores B para a realização de busca por títulos, e para ordenação de algumas categorias, principalmente por sua eficiência e simplicidade de implementação. Primeiramente, um código já existente, que possuía a inserção na árvore, foi obtido do site “Geeks for Geeks” e adaptado para suportar chaves com os elementos título do filme e posição deste no arquivo serial. Após, a inserção foi

modificada para comparar strings entre si como forma de determinar a ordem dentro de cada nodo, e uma função de busca foi criada para percorrer a árvore recursivamente procurando determinada string dentro dela. Porém, pela grande quantidade de filmes, houve um estouro da pilha de recursão, logo optou-se por fazer uma segunda versão da busca que usasse apenas iterações. Ao final, foi feito um laço para preencher cada uma das chaves com as informações dos arquivos e inseri-las na árvore B, além de funções para deletá-la, exibir os títulos em ordem alfabética e salvá-la em um arquivo binário. Fora isso, desejando-se explorar mais as funcionalidades da árvore B, produziram-se versões alternativas de todas as funções já existentes para lidar com chaves do tipo inteiro e ponto flutuante, possibilitando a ordenação de campos das películas como data de lançamento e popularidade.

2.3.2 Arquivos Invertidos

Consistindo em uma estrutura que inverte o modelo tradicional de indexação, o arquivo invertido apresenta uma lista de elementos que possuem o mesmo valor em um certo campo, permitindo filtrar um grande conjunto de elementos de acordo com os possíveis valores deste campo. Dessa forma, as principais aplicações deste tipo de arquivo de índice são para implementação de motores de busca e para realização de consultas eficientes, e por essas razões, dentro do projeto eles foram utilizados como meio para filtrar uma das características mais relevantes dos filmes, o gênero.

Para lidar com as listas dos arquivos invertidos, desenvolveu-se um esquema baseado nos sistemas FAT (File Allocation Table) de alocação e indexação de arquivos, que possui três áreas principais, dispostas uma abaixo da outra no arquivo, sendo elas: o boot, a tabela FAT em si e os blocos. A área de boot é responsável por conter informações como o offset do primeiro e último bloco, além da quantidade de bytes escritos no último bloco do diretório principal. Já na tabela FAT, cada elemento representa um par (bloco, valor), onde o segundo pode assumir quantias como 0x0000 (significando bloco livre), 0xFFFF (informando o fim de um encadeamento de blocos) e qualquer número entre 0x0000 e 0xFFFF, que representa o próximo bloco do encadeamento. Aproveitando-se do fato de que o tamanho da área de boot e as posições de cada elemento da tabela FAT são

constantes, utiliza-se o próprio offset do arquivo para informar a qual bloco aquela entrada da tabela se refere, significando, portanto, que a Tabela FAT guarda apenas o campo valor do par. Quanto aos blocos, cada um pode armazenar dois tipos de informação, representando um diretório ou um arquivo. No primeiro tipo, todo componente é uma 4-upla que simboliza uma categoria, cujos campos são: identificador da categoria, bloco inicial, bloco final do encadeamento e quantidade de bytes escritos no bloco final. Já no segundo tipo, são salvos os offsets dos filmes que pertencem àquela categoria.

2.4 Operações no Catálogo

Algumas das operações disponibilizadas pelo catálogo merecem destaque por serem as mais fundamentais e necessárias para o projeto. Um exemplo de uma delas é a busca por títulos, significativa para que o usuário obtenha facilmente as informações de cada película usando um dos principais aspectos delas: o nome. A pesquisa, como mencionado anteriormente, faz uso de uma árvore B, que por sua vez lida com chaves que guardam em si o título de cada filme e um ponteiro, na forma de um número inteiro que indica em posição em bytes, para o filme no arquivo binário principal, sendo preciso percorrê-la iterativamente comparando o conteúdo de cada chave com a string digitada pelo usuário para localizar a obra ou afirmar que ela não está presente. Fora isso, realizando-se um caminhamento nesta mesma árvore, tornou-se possível imprimir todos os títulos em ordem alfabética, e por possuir este fator de preservar uma ordem entre os registros, árvores B foram empregadas para todas as ordenações, resumidas em caminhamentos e paginação dos resultados destes.

Outra opção, a filtragem, fez uso dos arquivos invertidos, exibindo na tela de maneira paginada a lista correspondente a cada título do gênero selecionado, adquirida ao percorrer-se de maneira linear o encadeamento de blocos correspondente. O cálculo das estatísticas, como média de receita, orçamento, popularidade e total de votos por filmes, foi implementado com a ajuda de uma função que cria um vetor a partir da extração de um determinado campo de todos os elementos de uma mesma estrutura, servindo como base para o somatório da média. Já a inserção de dados de novos filmes foi feita com a manipulação do arquivo original, abrindo-o e inserindo ao seu final os novos dados, seguido de uma

deleção de algumas estruturas e arquivos de índice presentes (como para as árvores B) e a criação de novos com base nos arquivos atualizados.

2.5 Interface para o Usuário

Para que o usuário possa interagir com o catálogo, navegando dentre as suas opções, é necessário um meio que seja intuitivo, eficiente e acessível, facilitando a sua experiência e tornando-a aprazível. Tendo estas questões em mente e também prezando pela praticidade do projeto, adotou-se uma interface de linha de comando, onde, dentro da própria janela de execução do programa, o usuário pode manusear a aplicação ao digitar comandos previamente especificados por meio de texto na tela. Feita através de laços com switches para cada uma das operações disponíveis, o menu principal fornece ao usuário as seguintes possibilidades: pesquisar o título de um filme, filtrar ou ordenar as obras seguindo algum critério, apresentar estatísticas sobre as películas, inserir mais dados no arquivo original ou sair da aplicação.

Dentro das alternativas do menu existem ainda mais ações para serem tomadas, como na busca por títulos que permite, além de seu propósito original, exibir todos os filmes presentes no catálogo em ordem alfabética ou contrária a esta em forma de lista. Na filtragem, pode-se selecionar qualquer um dos 19 gêneros exibidos na tela, assim como visualizar de maneira paginada os títulos dos filmes em ordem alfabética referentes a cada um. Ao optar pela consulta de alguma estatística sobre as obras, pode-se observar as médias de receita, orçamento, popularidade e entre outras de todos os filmes tratados no catálogo.

Uma vez selecionada a opção de ordenação, um submenu é exibido, permitindo que o usuário decida entre visualizar todos os filmes organizados segundo seus títulos, data de lançamento ou popularidade, tanto na ordem crescente quanto decrescente. Escolhida alguma das alternativas anteriores, torna-se possível navegar entre as películas na forma de páginas com 20 obras cada, viabilizando-se avançar para a próxima, retornar para a anterior ou voltar para a escolha de ordenação. Já no caso da inclusão de novos arquivos com dados de produções cinematográficas, apenas pede-se ao usuário para que ele digite os nomes dos arquivos de filmes e títulos, e se estes estiverem corretos, as novas

informações são adicionadas aos dados originais. Ao pressionar a opção de sair do menu principal, uma mensagem de despedida é exibida e o programa encerra.

4 CONCLUSÃO

A partir do que foi exposto, pode-se perceber que o desenvolvimento de um catálogo de películas cinematográficas é um processo abrangente e volumoso, envolvendo desde a extração dos dados, seu tratamento, implementação das operações através de estruturas bem analisadas até a criação de uma interface de uso prático para o usuário. Devido às múltiplas etapas envolvidas, é necessário um planejamento estruturado e organizado entre os colaboradores do projeto, para que as variadas camadas da aplicação sejam feitas de forma gradual e constante, mantendo um ritmo adequado para cumprir todos os prazos. Além disso, todas as estruturas de dados empregadas devem ser criticamente estudadas para que se garanta eficiência e um bom nível de compreensão sobre o projeto.

Em relação à experiência dos desenvolvedores, notou-se que a quantia massiva de filmes aumentou consideravelmente a complexidade do catálogo, com várias estratégias, como criar um arquivo separado apenas para armazenar os títulos e adicionar ponteiros especiais nas estruturas, sendo aplicadas em nome da economia de espaço e tempo de processamento. Precisamente pelo número excessivo de obras, originou-se dentro do projeto um foco em escalabilidade, onde muitas vezes trabalhou-se com um conjunto menor de dados, mas sempre de maneira geral para que a mesma funcionalidade pudesse ser replicada para os demais. Em suma, apesar de se mostrar laboriosa e extensa, possuindo várias nuances e detalhes, a tarefa de construir um catálogo de filmes também foi satisfatória por se tratar de um tópico apreciado pelos criadores deste e pela esperança dos mesmos de que a aplicação tenha utilidade para usuários futuros.

REFERÊNCIAS

- ER Diagram. Disponível em: https://app.diagrams.net/. Acesso em: 23 maio 2025.
- HISTÓRIA do Cinema: da sua origem aos dias de hoje. Disponível em: https://www.aicinema.com.br/historia-do-cinema-da-sua-origem-aos-dias-de-hoje/. Acesso em: 12 jun. 2025.
- IMPLEMENTATION of B-Tree in C. Disponível em: https://www.geeksforgeeks.org/c/implementation-of-b-tree-in-c/. Acesso em: 06 jun. 2025.
- MOURA, Christopher. **Árvore B: O que é e para que serve?** Disponível em: https://medium.com/@ccmoura/%C3%A1rvore-b-o-que-%C3%A9-e-para-que-serve-71c949484527. Acesso em: 10 jun. 2025.
- SILVA, Eduardo da. **A API TMDB (The Movie Database API)**. Disponível em: https://eduardo-da-silva.github.io/aula-desenvolvimento-web/axios/tmdb-api. Acesso em: 17 maio 2025.
- TRAVIS BELL. **The Movie Database**. Disponível em: https://www.themoviedb.org/. Acesso em: 15 maio 2025.
- WRIGHT, Gavin. **O que é a Tabela de Alocação de Arquivos (FAT)?** Disponível em: https://www.techtarget.com/whatis/definition/file-allocation-table-FAT. Acesso em: 20 jun. 2025.