UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL- UFRGS
INSTITUTO DE INFORMÁTICA
CIÊNCIA E ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE CLASSIFICAÇÃO E PESQUISA DE DADOS
PROFESSOR LEANDRO KRUG WIVES

TRABALHO FINAL: ETAPA III

LUCIANO SAMPAIO DA SILVA
RICHARD MUNIZ ROSA

PORTO ALEGRE/RS
2023

Sumário

1 Descrição do Problema

O objetivo do trabalho consiste em desenvolver uma aplicação para solucionar a dificuldade dos usuários em encontrar filmes que atendam às suas preferências pessoais. A vasta quantidade de títulos e categorias disponíveis pode tornar a navegação por todas as opções desafiadora. Embora existam diversas fontes de informação sobre filmes, muitas delas não oferecem recursos avançados de pesquisa, o que dificulta a busca por filmes com base em critérios específicos, como gênero, produtora, país e década do filme.

Para solucionar esse problema, a aplicação a ser desenvolvida permitirá que o usuário pesquise em uma base de dados de filmes de acordo com critérios específicos, filtrando a busca e apresentando uma lista de filmes que atendam às suas preferências.

A aplicação será desenvolvida em Python e irá ler arquivos em disco para realizar as tarefas, o que tornará o processo de busca mais rápido e eficiente. A interface da aplicação será amigável e intuitiva, proporcionando uma experiência agradável para o usuário.

2 Funcionalidades Previstas

A aplicação que será desenvolvida terá diversas funcionalidades que permitirão ao usuário realizar consultas precisas na base de dados de filmes. O objetivo é facilitar a busca por títulos que atendam às suas preferências pessoais. Dentre as funcionalidades previstas, destacam-se:

1. Busca por nome de filme: o usuário poderá pesquisar por um título específico, o que facilitará a busca por filmes que ele já conhece e deseja assistir.
2. Busca por gênero: a aplicação permitirá que o usuário encontre filmes que pertencem a um determinado gênero, como comédia, drama, ação, romance, entre outros.
3. Busca por país: o usuário poderá pesquisar por filmes produzidos em um determinado país, o que possibilitará a descoberta de novos títulos de acordo com suas curiosidades culturais.
4. Busca por década: a aplicação permitirá que o usuário encontre filmes produzidos em uma determinada década, o que ajudará a descobrir títulos que foram populares em determinada época.
5. Mesclagem de consultas: o usuário poderá combinar diferentes critérios de busca, a fim de filtrar ainda mais os resultados e encontrar o filme ideal.

Com essas funcionalidades, a aplicação fornecerá uma solução prática e eficiente para ajudar os usuários a encontrar filmes que atendam às suas preferências, facilitando a navegação pela grande quantidade de títulos disponíveis na base de dados.

3 Cronograma de Atividades

Como forma de organizar as atividades entre os membros do grupo e subdividir as entregas em menores etapas, foi estipulado um cronograma de trabalho. A Tabela 1 apresenta as atividades observadas em primeiro momento e suas respectivas datas de entrega.

Tabela 1: Cronograma de atividades previsto para desenvolvimento do projeto.

Atividade	**28/Fevereiro**	**12/Março**	**31/Março**	**11/Abril**	**Responsável**
Delimitação do tema do projeto	X				Luciano/Richard
Definição de linguagem de programação para o projeto	X				Luciano/Richard
Busca da base de dados a ser utilizada no projeto	X				Richard
Refinar o problema e determinar as funcionalidades		X			Luciano/Richard
Criar o diagrama E/R para determinar o projeto de arquivos		X			Luciano
Entregar a Etapa II do projeto		X			Luciano/Richard
Filtrar dados relevantes no arquivo bruto do dataset		X			Richard
Gerar arquivos de entidades relevantes para a aplicação			X		Luciano
Gerar arquivos de relações entre as entidades			X		Luciano/Richard
Implementar funções de filtragem dos dados			X		Richard
Implementar interface de navegação para o usuário			X		Luciano
Implementar algoritmo de busca com estrutura de árvore				X	Luciano/Richard
Testes e aperfeiçoamentos do código				X	Luciano/Richard
Escrita do relatório	X	X	X	X	Luciano/Richard
Entrega e apresentação do projeto final				X	Luciano/Richard

4 Ferramentas e Bibliotecas

1. **csv**: Módulo Python para leitura, escrita e processamento de arquivos de texto csv.

   > Usado para fazer a filtragem inicial dos dados brutos tão como iniiciar o processamento inicial dos dados.

2. **json**: Módulo Python para processamento e conversão de strings json para objetos python (e vice-versa).

   > Usado pois algumas tabelas dos arquivos csv contém strings json, então esse módulo facilita o processamento dos mesmos.

3. **tkinter**: Módulo Python para desenvolvimento de interfaces gráficas

   > Usado para realizar a interface entre o usuário e o código, permitindo melhor visualização dos resultados.

4. **struct**: Módulo Python para processamento de bytes convertendo-os em tipos Python.

   > Usado para conversão de tipos primitivos de dados Python para dados no formato binário.

5. **Overleaf**: Ferramenta online para edição de documentos.

   > Usado para a criação do relatório deste trabalho.

6. **Docker**: Ferramenta de containerização e virtualização (semelhante a máquinas virtuais).

   > Usado para que o script/programa do projeto possa rodar em "qualquer"máquina independentemente de sistema operacional, configuração e versões de software.

7. **Github** Ferramenta de compatilhamento e versionamento de código.

   > Usado para que os integrantes do projeto possam compartilhar códigos feitos e terem um ponto central para manter/obter o código atualizado.

5 Projeto de Arquivos

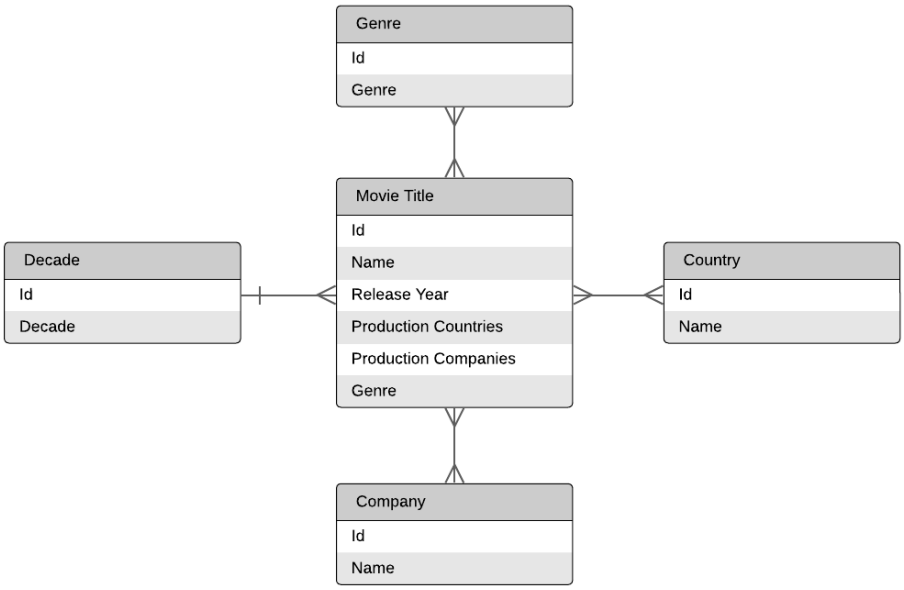

Figura 1: Diagrama E/R proposto para o projeto.
Fonte: Autoria Própria.

6 Desenvolvimento

6.1 Busca por títulos - Arvore Trie

Para realizar a busca por títulos, foi desenvolvida uma árvore Trie, com intuito de indexar e pesquisar por palavras em um conjunto de dados de filmes. A Trie é uma estrutura de dados eficiente para armazenar e buscar palavras em tempo linear em relação ao tamanho da palavra, o que a torna uma escolha adequada para a tarefa de indexação de palavras em filmes.

A classe *TrieNode* define os nós da Trie, que contêm informações sobre as palavras armazenadas na Trie. Cada nó tem uma lista de filhos que representam as letras das palavras, um indicador de fim de palavra para marcar o final de uma palavra e uma lista de filmes associados a esse nó.

A classe Trie define a estrutura da árvore e as funções de inserção e pesquisa. A função *insert* percorre cada letra da palavra a ser inserida e adiciona um novo nó filho para cada letra, se necessário. Ao final da palavra, o indicador de fim de palavra é definido como *True* e o filme associado à palavra é adicionado à lista de filmes do nó.

A função *search* percorre a Trie a partir do nó raiz, verificando cada letra da palavra de consulta para ver se ela tem um nó filho correspondente. Se a letra não estiver presente, a pesquisa termina e uma lista vazia é retornada. Se a letra estiver presente, o nó filho correspondente é visitado e a pesquisa continua até que a palavra seja encontrada ou não exista um nó filho correspondente. Após uma busca ser realizada, o resultado é exibido ao usuário em ordem alfabética pelo nome do filme, utilizando a função de ordenamento nativa do *python*.

As funções *save trie* e *load trie* são usadas para salvar a Trie em um arquivo e carregá-la de volta a partir do arquivo, respectivamente. Isso permite que a Trie seja reutilizada posteriormente sem ter que ser construída novamente.

6.2 Interface gráfica

O código apresentado é uma interface gráfica construída utilizando a biblioteca Tkinter do Python. Essa interface é destinada à pesquisa de filmes, permitindo ao usuário inserir informações sobre título, década de lançamento, gênero, país de origem e produtora.

O Tkinter é uma biblioteca padrão em Python para criação de interfaces gráficas do usuário (GUIs). Ele fornece uma maneira fácil de criar interfaces de usuário interativas e responsivas para aplicativos Python. A biblioteca fornece vários widgets de interface do usuário, como botões, caixas de texto, rótulos, menus suspensos entre outros. Esses widgets podem ser organizados em janelas, quadros e outros contêineres para criar uma interface do usuário funcional.

Os recursos utilizados neste projeto foram relacionados ao suporte a eventos de interface do usuário, sendo por exemplo a seleção de opções através de caixas suspensas, botões, entrada

e saída de texto. Esses eventos podem ser manipulados por meio de funções de retorno de chamada, permitindo que o programa responda às ações do usuário de maneira apropriada.

Ao clicar no botão de pesquisa, a interface invoca a função *show result*, que retorna uma lista de filmes que satisfazem os critérios de busca informados pelo usuário. Essa lista é exibida na área de resultados, que consiste em um widget de texto com uma barra de rolagem. Além disso, a interface possui um botão de "Reversão"que permite inverter a ordem dos resultados exibidos, e um botão "Carregar CSV"que permite ao usuário carregar um arquivo CSV contendo informações de filmes para pesquisa.

7 Guia do Usuário

8 Considerações finais

8.1 Busca por títulos - Arvore TRIE

Com intuito de medir o tempo para que o código leva para armazenar aproximadamente 4000 títulos de filmes na estrutura Trie, foi adicionado um temporizador auxiliar, o qual retornou aproximadamente 0,52 segundos. Uma vez que a Trie é construída e salva em um arquivo, o tempo de busca de filmes que correspondem a uma consulta do usuário é muito rápido. O uso da biblioteca pickle para serializar e deserializar a Trie também ajuda a otimizar o desempenho do código, permitindo que a Trie seja facilmente salva e carregada a partir de um arquivo.

Inicialmente ao desenvolvimento do código, foi considerado utilizar oa algorítmo *Bubble Sort* para realizar o ordenamento dos filmes após a busca. Contudo, o algorítmo apresentou um tempo de execução muito maior do que o desejado. Ao realizar a pesquisa da palavra "The", que é comum a muitos filmes, o algorítmo levou 17,34 segundos para exibir o retorno. Como forma de otimizar este tempo, a ordenação foi realizada pela função *Quick Sort built-in* de ordenamento do Python, reduzindo o tempo da busca da mesma palavra para 6,77 segundos.

O desenvolvimento desta funcionalidade não apresentou grande complexidade, considerando que a linguagem utilizada possui diversos recursos que facilitam seu desenvolvimento.

8.2 Interface gráfica

Inicialmente, o desenvolvimento de uma GUI usando o pacote tkinter foi desafiador para o grupo, uma vez que a equipe não dominava completamente suas funcionalidades. No entanto, com auxilio da bibliografia (MOORE, 2018), foi possível compreender melhor o pacote e realizar a interface apresentada na Figura 2.

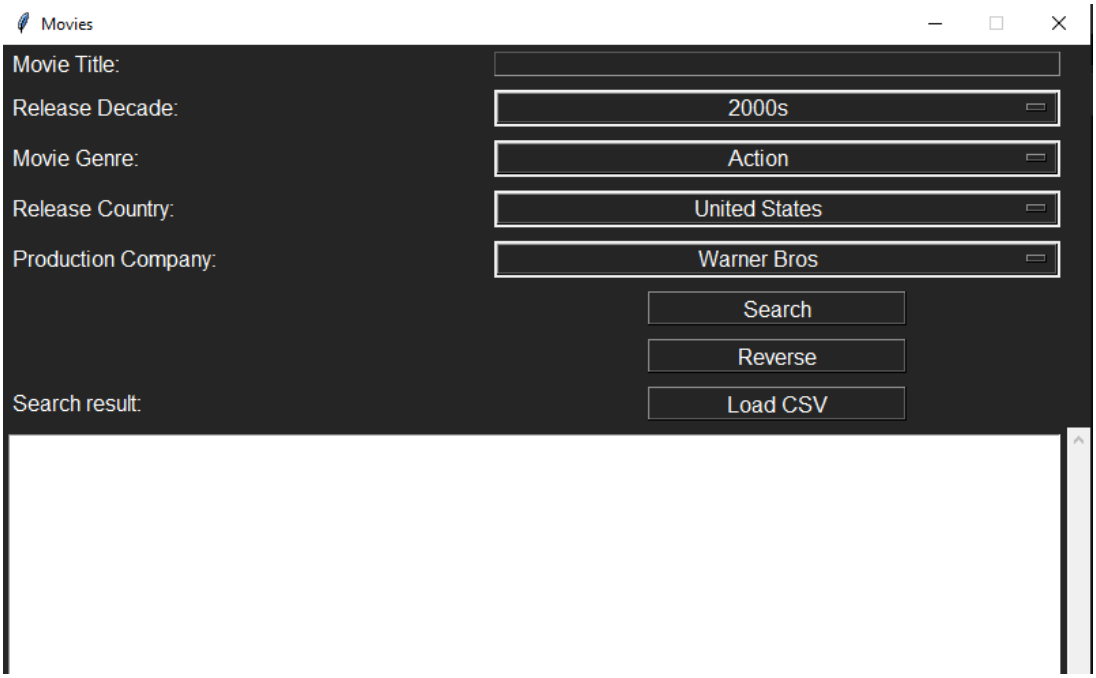

Figura 2: Interface gráfica realizada utilizando o pacote Tkinter.
Fonte: Autoria Própria.

Referências

MOORE, A. D. **Python GUI programming with Tkinter**. [s.n.], 2018. Disponível em: <https://books.google.com.br/books?id=2kBbDwAAQBAJ&lpg=PP1&ots=NfxjRTPQsi&dq=tkinter\%20with\%20python&lr&pg=PP3#v=onepage&q&f=true>.