Projecto de Algoritmos e Técnicas de Programação
Relatório Final
Grupo 18

Ema Martins, a97678 Luís Gonçalves, a95637
Henrique Malheiro, a97455

2022-01-20

Conteúdo

1 Introdução

No âmbito da UC de Algoritmos e Técnicas de Programação, foi proposto aos alunos do 2º ano de Engenharia Biomédica criar um programa para operar uma base de dados de filmes. Este relatório descreve o processo de criação, desenvolvimento e resolução de problemas até à conclusão do programa.

O relatório será dividido em 6 partes. A "1. Introdução", "2. Ambiente de Trabalho", "3. Análise e Especificação", "4. Concepção/Desenho da Resolução", "5. Codificação e Testes" e as "6. Conclusões". A segunda secção é caracterizada pela descrição do equipamento e aplicações utilizadas. A terceira servirá para descrever o problema com maior profundidade. Na seguinte, abordaremos as soluções para os problemas encontrados na secção anterior. Na quinta, explicaremos em detalhe as soluções encontradas. Terminaremos com as conclusões deste projeto.

2 Ambiente de Trabalho

Na criação deste programa, trabalhamos com o sistema operativo Windows de versão 10. Utilizou-se o compilador "Visual Studio Code" na versão 1.63 acompanhado do Python 3.9. Foram instaladas algumas extensões do Python, sendo elas "matplotlib" 3.5.0 e o "PySimpleGUI" 4.55.1.

3 Análise e Especificação

O programa manipula uma base de dados com diversos filmes em que cada um tem associado um título, um ano de lançamento, um elenco, a/as categorias e um id. Com estes dados podemos realizar as funções base do programa.

Lista de funções:

-Carregar a BD;

-Guardar a BD;

-Inserir um novo filme na BD;

-Consultar um filme na BD (a partir do id ou a partir do título);

-Listar todos os filmes por ordem alfabética de título;

-Listar todos os filmes de um determinado género;

-Listar todos os filmes de um determinado ator;

-Alterar um registo na BD;

Parâmetros estatísticos:

-Número de filmes da BD;

-Lista de atores ordenada alfabeticamente em que, cada ator, tem a ele associada uma lista dos

títulos e ids dos filmes em que participou;

-Lista de géneros de filmes ordenada alfabeticamente em que, cada género, tem a ele associada uma lista dos títulos e ids dos filmes que pertencem a esse género.

De forma a ser mais intuitivo para o utilizador, estas funcionalidades foram incorporadas numa interface gráfica.

4 Concepção/desenho da Resolução

O programa está dividido em 2 ficheiros python. O "Projeto.py" contém as funções que são usadas no "Projeto_Visuals.py" que inclui o código da interface gráfica.

4.1 Estruturas de Dados

O modelo da base de dados é uma lista de dicionários, em que este último é composto por cinco chaves ("title","year","cast","genres","id") com os seus respetivos valores. No caso das chaves "cast" e "genres" terem mais do que um ator/género, os seus valores serão uma lista.

4.2 Algoritmos Projeto.py

CarregarBD: Tem como parâmetro "fnome"(diretoria da BD), permitindo assim abrir o ficheiro json dessa diretoria.

InserirID: Percorre a BD, adicionando um ID a cada filme.

GuardarBD: Cria um novo ficheiro de nome "fnome"(parâmetro) e posteriormente guarda-o em json.

InserirBD: Cria um novo filme adicionando-o à BD.

ProcurarID: Percorre a BD até encontrar o filme com o ID pretendido. Devolve a entrada completa.

ProcurarTitulo: Dado um argumento, título ou uma parte dele, percorre a BD e retorna todos os filmes que contenham esse argumento.

OrdenarTitulo: Ordena a BD por ordem alfabética dos títulos, através da função *chaveOrd*.

ProcurarGenero: Dado um argumento, género, percorre a BD e retorna todos os filmes o contenham.

ProcurarAtor: Dado um argumento, nome do ator ou uma parte dele, percorre a BD e retorna todos os filmes que contenham esse argumento.

QuantosFilmes: Retorna o número de filmes da BD.

DistribPorGenero: Retorna um dicionário em que as chaves são os diversos géneros de filmes e os valores o número de filmes desse género. Se esse valor for menor do que 1% do número

de filmes da BD, irão ser adicionados ao valor da chave "Others".

PlotDistribPorGenero: Através do módulo "matplotlib", faz um gráfico circular dos géneros com os valores obtidos na distribPorGenero.

DistribPorAtor: Cria um dicionário com todos os atores e o número de filmes em que participou. Retorna um segundo dicionário com os 10 atores com mais participações.

PlotDistribPorAtor: Através do módulo "matplotlib", faz um gráfico circular do top 10 de atores com mais participações em filmes, com os valores obtidos na distribPorAtor.

ListaAtor: Cria uma lista com todos os atores da BD, e organizando-os alfabeticamente. Percorre a BD e a lista com os atores simultaneamente. Se o nome do ator aparecer em ambos, adiciona a um dicionário, na chave com o nome do ator, uma lista com os IDs e com os títulos dos filmes em que participou. Retorna este último dicionário.

ListaGen: Cria uma lista com todos os géneros da BD, e organizando-os alfabeticamente. Percorre a BD e a lista com os géneros simultaneamente. Se o género aparecer em ambos, adiciona a um dicionário, na chave com o nome do género, uma lista com os IDs e com os títulos dos filmes que dessa categoria. Retorna este último dicionário.

AlterarTitulo: Para um certo ID, troca o título do filme pelo "novo" (parâmetro).

AlterarAno: Para um certo ID, troca o ano do filme pelo "novo" (parâmetro).

AlterarAtores: Para um certo ID, troca um ator do filme, "velho" (parâmetro), pelo "novo" (parâmetro).

AlterarGenero: Para um certo ID, troca um género do filme, "velho" (parâmetro), pelo "novo" (parâmetro).

Ord2: Ordena a BD por ID

4.3 Algoritmos Projeto_Visual.py

Neste ficheiro utilizou-se o módulo "PySimpleGUI" e as suas diversas funções.

A interface principal é composta por duas colunas, separadas por um divisor vertical. A primeira coluna contém os vários botões que, após serem selecionados, originam eventos. Estes provocam a alteração da segunda coluna, "painel de dados".

A segunda coluna tem sempre os elementos de input e output necessários para o funcionamento das funções referidas no ponto anterior. Alguns destes elementos são botões, texto, tabelas, quadros, as listbox e o elemento input.

A maioria destes elementos estão invisíveis através da propriedade visible=False. A alteração desta condição é a base do funcionamento da interface gráfica.

Utilizamos outras propriedades, como é exemplo tooltip, fontes, tamanhos, chaves, justificação, do_not_clear e horizontal_scroll. Para as tabelas, headings, vertical_scroll_only e values. Tivemos ainda em consideração propriedades visuais, como exemplo os temas e as cores para as tabelas (background_color, alternating_row_color, texto_color).

5 Codificação e Testes

No inicio da fase de testagem, achamos que os erros seriam provenientes duma errada utilização por parte do utilizador, no "Projeto_Visuals.py".

Os principais erros encontrados foram:

Valores inteiros: O ano e o Id devem ser sempre valores inteiros, então sempre que necessário a introdução dos mesmos tivemos que ter atenção aos ValueError, abrindo um popup com uma mensagem de erro.

Inserir Atores e Géneros: No evento Inserir, quando o novo filme tem mais do que um Ator ou Género, os mesmos têm que ser separados por vírgulas, de maneira a que seja criada uma lista com os mesmos. Para prevenir uma errada utilização, pusemos uma mensagem ToolTip que vai indicar a maneira correta de utilizar esta função.

Erro do PySimpleGui nas tabelas e frames: Sempre que estes passavam a visible=False, ficavam sempre uma imagem deles. Para resolver este problema, separou-se em 3 colunas.

Esta solução criou um segundo erro, levando a que o nome dos eventos nas colunas não sejam fixos. Assim, criamos uma chave única para cada botão de cada coluna. Inputs incompletos: A ausência de caracteres nos inputs, levava a que o programa não trabalhasse. Verificamos então se algum input estaria vazio, abrindo um popup indicando o problema.

ID não existe na BD: No caso do ID, apesar de numérico, ser maior do que o tamanho da BD ou até negativo o programa não iria funcionar. Limitamos então o id de zero até ao tamanho da BD.

Erro na troca de Ator/Género: Ao alterar um ator ou género, é necessário seleciona-lo na listbox. Se não for selecionado, surgirá um popup com uma mensagem de erro.

Erros Python: Existiam erros tais como *ValueError*, *TypeError* e *IndexError*. Resolvemo-los através da estrutura try e except.

Diretoria da BD errada: Se a diretoria da BD estiver errada, abrirá um popup de erro.

Ao testar a aplicação, encontramos maneiras de a simplificar e melhorar, tais como mensagens de conclusão com sucesso de cada evento, dicas de como operar a aplicação e algumas simplificações dos algoritmos.

6 Conclusões

Neste relatório, começamos por introduzir e analisar o problema de forma crítica, de maneira a tentar resolvê-lo de uma forma simples e eficaz. De seguida, analisamos a estrutura do projeto, bem como o funcionamento de cada função utilizada. Terminamos por explicar os diversos erros que nos foram surgindo ao longo do trabalho e a maneira como os resolvemos.

O projeto inclui todas as funcionalidades propostas e as três funcionalidades adicionais.

Apesar de a aplicação não ter ficado totalmente otimizada, ficamos satisfeitos com o resultado.