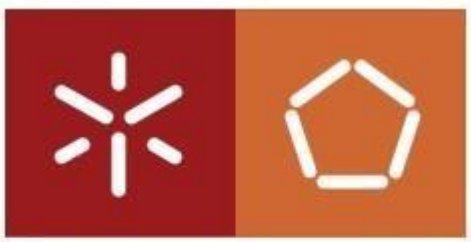
Projeto Laboratórios de Informática III

Fase I

Projeto desenvolvido por:
Pedro Teixeira(A103998), Jorge Fernandes(A104168), Diogo Fernandes(A104260)

Grupo 54

Licenciatura em Engenharia Informática



Universidade do Minho
Escola de Engenharia

Departamento de Informática
Universidade do Minho

Conteúdo

Capítulo 1
Introdução

No âmbito da unidade curricular de Laboratórios de Informática III, do ano 2024/2025, foi-nos proposto o desenvolvimento deste projeto no qual temos de implementar uma base de dados em memória que armazene dados fornecidos pelos docentes. Com este trabalho pretende-se dar a conhecer aos alunos os princípios fundamentais da Engenharia de Software e da linguagem C, nomeadamente modularidade e encapsulamento, estruturas dinâmicas de dados, validação funcional, e medição de desempenho (computacional, consumo de memória, etc), compilação, linkagem, definição de objetivos de projeto com base nas suas dependências, depuração de erros, avaliação de desempenho e consumo de recursos, e gestão de repositórios colaborativos.
O projeto está dividido em duas fases, e esta primeira fase consiste em implementar o parsing dos dados e o modo *batch*(programa-principal). Este modo consiste em executar várias *queries* sobre os dados de forma sequencial, estando esses pedidos armazenados num ficheiro de texto cujo caminho é recebido como argumento.
Assim, para além da leitura e interpretação dos dados dos ficheiros, este processo requer também o armazenamento dos mesmo em estruturas de dados versáteis e de rápido acesso.

Capítulo 2

Desenvolvimento

Para uma maior facilidade na execução e organização do código, o nosso grupo começou por organizar e planear. Assim, conseguimos uma visão mais clara daquilo que temos e queremos fazer, para que consigamos evitar ter de refazer grandes partes de código devido a implementações ineficientes ou menos práticas. Começamos por imaginar uma arquitetura de como seria a ligação entre os diferentes módulos e a forma como iriam comunicar entre si. Tendo em consideração o mencionado, chegamos à seguinte arquitetura:

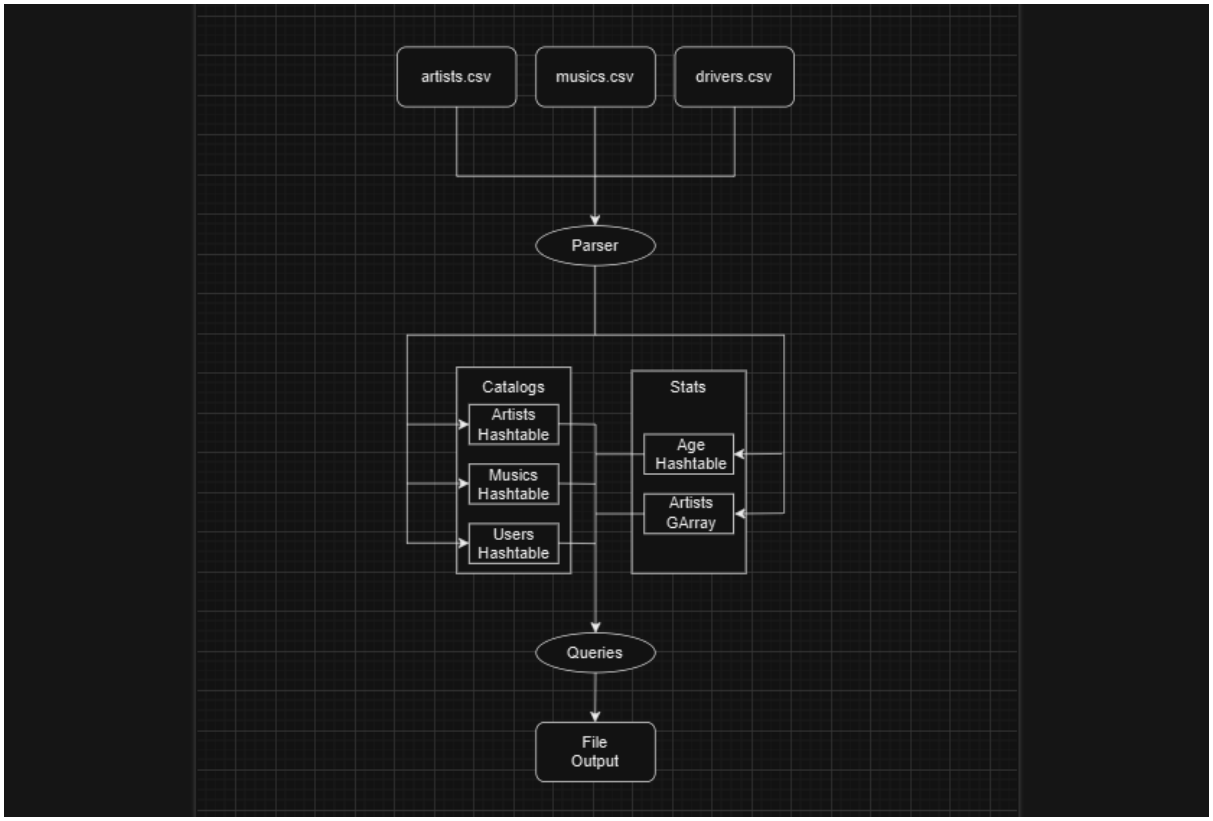


Figura 1: Diagrama da estrutura utilizada para a 1ª fase

Começamos por fazer um módulo de *parsing* geral dos ficheiros *.csv* e a verificação das entidades neles incluídos (*artistas*, *músicas* e *utilizadores)*. Concluímos que *hash tables* seriam o ideal para armazenar os dados criando, portanto, três *hash tables* – uma para cada entidade.

QUERIES

Query 1:
A *query* 1 consiste em listar o resumo de um utilizador, consoante o identificador recebido por argumento. A *query* recebe como argumento o identificador único do utilizador.
Posto isto, esta *query* foi a mais simples devido ao facto deste identificador único ser a chave de acesso a cada utilizador inserido na *hash table*, logo, foi possível usar a função *lookup*(função predefinida da *glib*) para facilmente aceder ao utilizador e às suas informações. Toda a informação impressa já estava armazenada na *hash table* o que mais uma vez contribuiu para a facilidade na resolução desta *query*.

Query 2:
A *query* 2 consiste em listar os top N artistas com maior discografia. A *query* recebe como argumento o número de artistas que devem constar do output, podendo ainda receber (ou não) o filtro de país, sendo que quando presente só devem ser considerados artistas desse país.
Com vista a facilitar a execução desta *query*, foi criado um atributo "*Discography Time*" na entidade artistas e utilizada a componente *stats*, onde ao longo do *parsing* foram armazenados num *GArray* os artistas. Isto ajudou muito no processo da *query* 2 devido à facilidade de organizar um *GArray* tendo em conta as funções predefinidas da *glib*.

Query 3:
A *query* 3 consiste em listar os géneros de música mais populares numa determinada faixa etária. A *query* recebe como argumento as idades mínima e máxima da faixa etária.
Para facilitar a execução desta *query* foi utilizada a componente "*stats*" e criada uma estrutura "Género Musical". A informação sobre os *likes* por idade para cada género é armazenada nas "*stats*", e na *query* é usada para adquirir o número total de *likes* para os parâmetros de idade pedidos. Posteriormente, este número total de *likes* é associado ao género através da estrutura referida anteriormente, estrutura esta que é inserida num *GArray* para facilmente ser organizada para obter o output desejado.

Capítulo 3

Dificuldades Sentidas

Durante o desenvolvimento deste projeto, diversas dificuldades foram enfrentadas, afetando diferentes áreas do processo de desenvolvimento. As dificuldades surgiram na implementação das estruturas de dados e bibliotecas recentemente aprendidas devido à sua complexidade e à sua diferenciação. Além disso, enfrentamos dificuldades específicas na deteção e correção de *memory leaks*, o que exigiu atenção extra relativamente à gestão manual de memória.

Capítulo 4

Desempenho e Testes

Este capítulo detalha o processo de teste realizado no projeto, focando nos conjuntos de dados. Os testes desempenham um papel crucial no desenvolvimento de software, garantindo a qualidade e viabilidade do sistema. Neste contexto, realizamos testes abrangentes nos conjuntos de dados para validar o desempenho, a precisão e a escalabilidade.

	Tempo de execução	Memória gasta
Teste 1 (ASUS Vivobook 16)	6.614555 seg	375072 KB
Teste 2 (HUAWEI Matebook D14)	15.927062 seg	374912 KB
Teste 3 (LENOVO Thinkpad i7-9th gen)	8.877878 seg	375168 KB

Capítulo 5

Conclusão

De um modo geral, na primeira fase do desenvolvimento do projeto, o foco principal estava em garantir que o código desenvolvesse fortes requisitos de encapsulamento e modularidade, assim como os critérios de organização e reutilização de código. Dada a complexidade e a extensão do projeto, era crucial fomentar a modularidade para separar as responsabilidades e simplificar o desenvolvimento contínuo e a expansão futura do projeto. A par disso, garantir uma gestão adequada da memória, livre de *memory leaks*, era imprescindível para melhor eficiência do programa. Esses objetivos iniciais, embora desafiadores, são passos cruciais para estabelecer uma base sólida para a continuidade do projeto.

Assim, os conceitos de encapsulamento, modularidade e gestão de memória continuarão a guiar o desenvolvimento nas próximas etapas, consolidando as fundações para um sistema robusto e eficiente.