



Universidade do Minho
Escola de Engenharia

Relatório 1ª Fase | Laboratórios de Informática III

Grupo 34 | 2024/2025

- Afonso Martins (A106931)
- Gonçalo Castro (A107337)
- Luís Felício (A106913)

Índice

Relatório 1ª Fase Laboratórios de Informática III	1
Grupo 34 2024/2025	1
1. Introdução	3
2. Desenvolvimento	4
2.1. Estrutura Aplicada	4
2.2. Funcionamento do Projeto	5
2.3. <i>Queries</i>	8
2.3.1. <i>Query 1</i>	8
2.3.2. <i>Query 2</i>	8
2.3.3. <i>Query 3</i>	8
2.4. Otimizações e Limitações	9
2.5. Dificuldades Sentidas	9
3. Conclusão	10

1. Introdução

O presente relatório tem como objetivo apresentar informações relativas à 1ª Fase do Projeto Final da Unidade Curricular de Laboratórios de Informática III, pertencente ao 2º Ano da Licenciatura em Engenharia Informática, realizada no ano letivo 2024/2025, na Universidade do Minho.

O projeto final referido consiste na implementação de uma base de dados em memória, utilizando-se ficheiros .csv fornecidos pelos professores, constituídos por dados relativos a um Serviço de música (sendo as principais entidades: **musicas**, **utilizadores**, **artistas**). O objetivo principal centra-se então em armazenar dados relativos a estas entidades e implementar métodos de pesquisa e associações entre estes.

São aplicados neste trabalho conceitos fundamentais como modularidade, encapsulamento, reutilização e abstração de dados, entre outros...

Na 1ª fase agora concluída, era necessário implementar o *parsing* e validação dos dados de entrada, bem como a realização de 3 *queries* definidas pela equipa docente.

2. Desenvolvimento

No projeto atual foi implementado o *parsing* de dados de entrada, sendo possível executar as *queries* sobre os dados sequencialmente, encontrando-se esses pedidos guardados num ficheiro de texto cujo caminho é recebido como argumento. No caso de se encontrar alguma linha do *Dataset* que não esteja em conformidade com os requisitos de cada parâmetro esta deverá ser adicionada a um ficheiro de erros adequado a cada Tipo/Entidade.

Assim, o programa requer não só a leitura e interpretação dos dados presentes nos ficheiros, como o seu respetivo armazenamento em estruturas de dados flexíveis e de eficiente acesso.

Para tal, foi permitida a utilização da biblioteca ‘*glib*’, fornecendo implementações pré-definidas de estruturas de dados úteis e otimizadas (*Hashtables*, Listas, etc.).

2.1. Estrutura Aplicada

No desenvolvimento deste projeto, utilizou-se a estrutura geral proposta pelos docentes, incluindo sempre a nossa interpretação do problema.

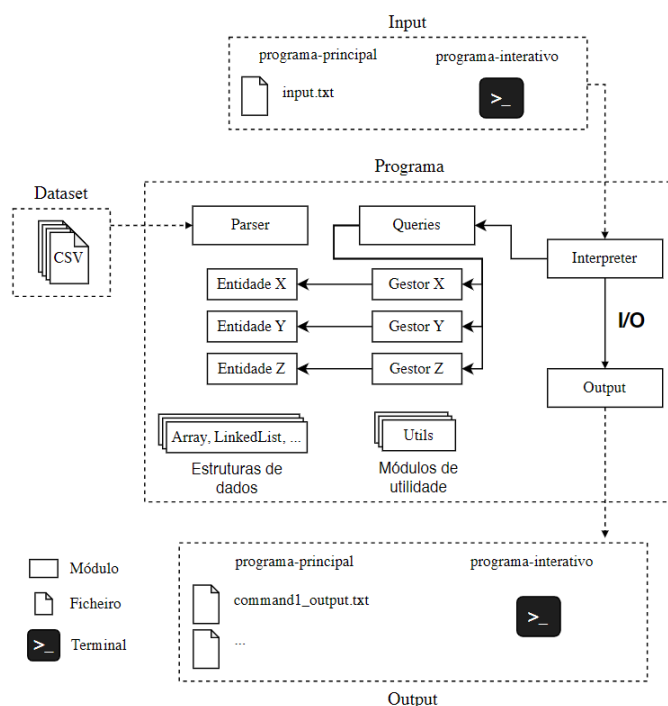


Figure 1: Estrutura geral fornecida pelos Professores

2.2. Funcionamento do Projeto

Para compilar e executar o nosso projeto é necessário seguir estes passos:

Limpar Compilação Anterior

```
make clean
```

Este comando elimina quaisquer ficheiros temporários e binários previamente compilados.

Compilar o Projeto

```
make
```

Este comando irá:

- Compilar todos os módulos do projeto, isto é todos os ficheiros `.c` em `src/`
- Gerar dois executáveis: `programa-principal` e `programa-testes`

Executar o Programa Principal

Após a compilação bem-sucedida, pode executar o programa principal, que irá executar os comandos e as respetivas *queries*, escrevendo os resultados na pasta `resultados/`. Para isso, execute o seguinte comando:

```
./programa-principal <caminho_para_dataset>  
<caminho_para_ficheiro_de_input>
```

Executar o Programa de Testes

Alternativamente ao programa-principal, pode executar o programa de testes. Este compara os outputs gerados com os resultados esperados. Para isso, execute o seguinte comando:

```
./programa-testes <caminho_para_dataset>  
<caminho_para_ficheiro_de_input>  
<caminho_para_resultados_esperados>
```

A primeira fase do processamento envolve o *Parser*, um módulo responsável por interpretar e converter os dados do *Dataset* para um formato que possa ser manipulado internamente pelo programa, geralmente em estruturas de dados como as *HashTables*. O *Parser* prepara os dados para serem validados e armazenados de forma adequada. Além disso este módulo também é responsável por ler e decodificar os comandos provenientes do ficheiro de Input recebido como parâmetro.

Após o parsing, os dados passam por um processo de Validação. Durante esta etapa, o programa verifica se os dados extraídos estão corretos e cumprem os requisitos estabelecidos no ENunciado, como tipos de dados adequados e valores dentro de intervalos aceitáveis. Se forem encontrados erros, como linhas mal formatadas ou dados inválidos, essas entradas são copiadas para um ficheiro de erros respetivo a cada uma das três Entidades, permitindo que os dados corretos sigam para a próxima fase.

Os dados validados são então armazenados numa Base de Dados em Memória, que é estruturada em três entidades *HashTables* principais destinadas a cada uma das Entidades/Tipos anteriormente descritos. Estas entidades são geridas por módulos dedicados, denominados Gestores, que garantem uma organização por Tipos de funções que controlam as estruturas de dados a eles associados.

O projeto também recebe comandos através de um ficheiro de *Input*. Esses comandos definem as consultas, ou *Queries*, que devem ser realizadas sobre os dados armazenados na base de dados. As *Queries* são instruções específicas que solicitam informações detalhadas, procuras e filtragem sobre os dados existentes.

Quando o sistema recebe uma consulta, o módulo *Queries* é responsável por processá-la, interagir com a base de dados e extrair as informações solicitadas. O resultado dessa interação é então guardado num ficheiro de Output diferente para cada comando (ou seja um ficheiro separado para cada invocação de uma *Querye*), que contém as respostas às procuras realizadas.

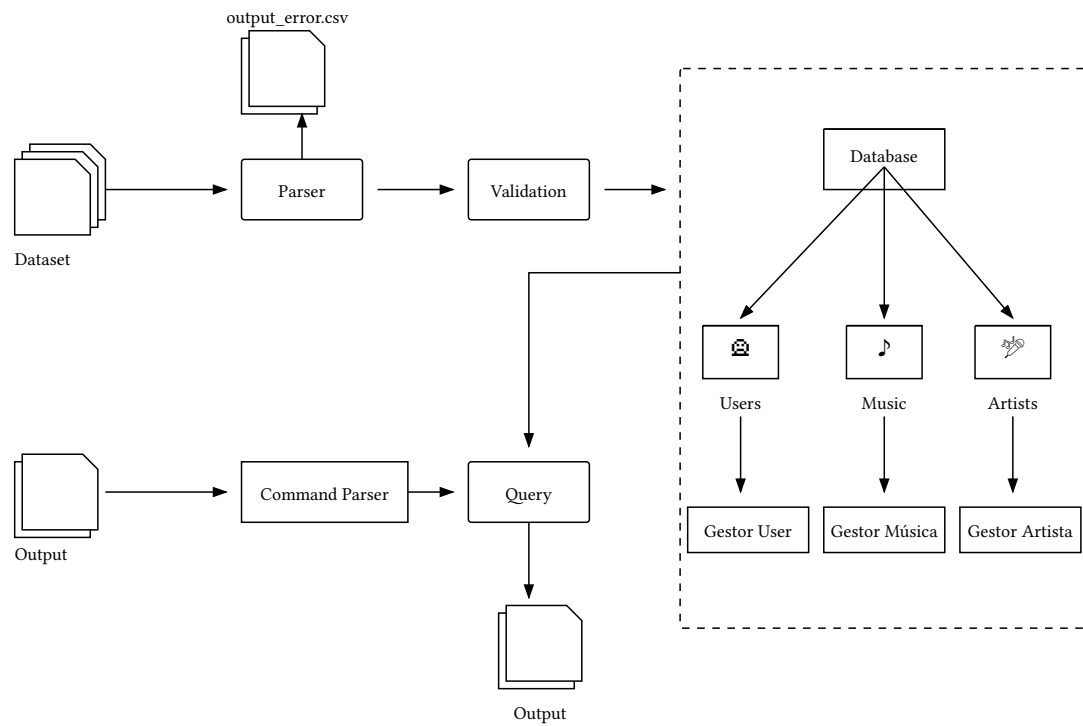


Figure 2: Estrutura Final da 1ª Fase do Projeto

2.3. *Queries*

Foram apresentadas 3 *Queries* no âmbito do Enunciado fornecido, que foram implementadas da forma de seguida descrita:

2.3.1. *Query 1*

A **Query 1** tem como objetivo fornecer um resumo de um utilizador específico com base no seu identificador único que é fornecido como argumento. Esta deve escrever no ficheiro de saída os seguintes dados do utilizador: email, primeiro nome, último nome, idade e país do utilizador. Caso o identificador fornecido não seja válido ou não seja encontrado na lista de utilizadores, uma linha vazia deve ser retornada.

2.3.2. *Query 2*

A **Query 2** tem como objetivo listar os top N artistas com a maior discografia, sendo que N é um argumento fornecido e, opcionalmente, pode ser fornecido também um filtro de país. No caso de existir este filtro “país”, serão considerados apenas os artistas desse país específico. Os dados que devem ser escritos no ficheiro de saída são: o nome, o tipo (individual ou coletivo), a duração total da discografia (igual à soma da duração de todas as músicas associadas a esse mesmo artista) e o país do artista. Artistas individuais e coletivos são tratados de forma separada e, em caso de empate na duração da discografia, a escolha é feita pela ordem crescente dos identificadores.

2.3.3. *Query 3*

A **Query 3** tem como objetivo identificar quais são os géneros de música mais populares de uma determinada faixa etária, definida pelos argumentos de idade mínima e máxima dadas no *input* do comando. O output dessa *query* é uma lista ordenada de géneros de música em ordem decrescente de popularidade, juntamente com o número total de *likes* associados a cada género. A popularidade de um dado género é determinada pela quantidade de *likes* das músicas desse género, considerando apenas os utilizadores que se encontram dentro da faixa etária especificada. Em caso de empate no número de likes, o desempate é feito pela ordem alfabética dos géneros.

2.4. Otimizações e Limitações

As únicas possíveis otimizações pensadas a serem futuramente implementadas são diminuição de memória utilizada e do tempo de execução. No que toca à diminuição de memória utilizada, poderíamos ter removido alguns dados nas Estruturas de Dados utilizadas (que não seriam úteis para os outputs) e pensar noutras maneiras de fazer as funções necessárias de uma forma mais otimizada e preocupada pela conservação de memória.

2.5. Dificuldades Sentidas

Das principais dificuldades que tivemos foi diminuir os *leaks* ao nível da memória de forma a passar na plataforma de testes com um bom resultado. Usando ferramentas como o *Valgrind* e o *GDB* estes entraves foram superados com bastante sucesso.

Na *query* 3 tivemos as principais dificuldades com vários erros, tais como *segmentation fault*, *malloc consolidate*, etc... Depois de serem ultrapassadas essas dificuldades, deparamo-nos com outro problema: o número de likes e consequentemente a ordem de géneros de música. Este problema estava relacionado com alguns erros de formatação dos parâmetros retirados dos ficheiros .CSV. Esse problema foi facilmente ultrapassado depois de ser descoberto recorrendo a técnicas de *Debug*, fazendo uso novamente das ferramentas descritas no parágrafo anterior.

3. Conclusão

A primeira fase deste projeto permitiu-nos aprofundar o conhecimento sobre a manipulação e organização de dados em memória, bem como a sua correta validação e interpretação, utilizando como base ficheiros CSV. Através da implementação das queries propostas, foi possível aplicar conceitos fundamentais de programação, como a modularidade, encapsulamento e reutilização de código, garantindo a correta separação de responsabilidades dentro do sistema apresentado e desenvolvido.

Além disso, o uso de ferramentas como o *Valgrind* e o *GDB* para deteção e correção de erros de memória foi essencial para garantir a eficiência e a robustez do programa, o que reforçou a importância das boas práticas no desenvolvimento de *software*.

Apesar dos desafios encontrados, especialmente na implementação da *Query 3* e na otimização do consumo de memória, conseguimos superar esses obstáculos, alcançando um resultado funcional, de acordo com os requisitos iniciais. Esta primeira fase forneceu uma base sólida para a continuidade do projeto, sendo que os próximos passos se deverão passar pela melhoria de performance e também no alargamento das funcionalidades desenvolvidas.

Em suma, a realização desta fase inicial permitiu consolidar conhecimentos teóricos e práticos, sendo fundamental para o sucesso das etapas subsequentes do projeto e para a nossa aprendizagem como um todo.