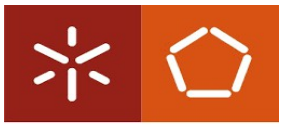


Universidade do Minho

Relatório 1ª Fase | Laboratórios de Informática III | Grupo 37 | 2024/2025

Francisco Lage (A106813) **Ricardo Neves (A106850)** **Rui Faria (A106899)**

Índice

1.Introdução

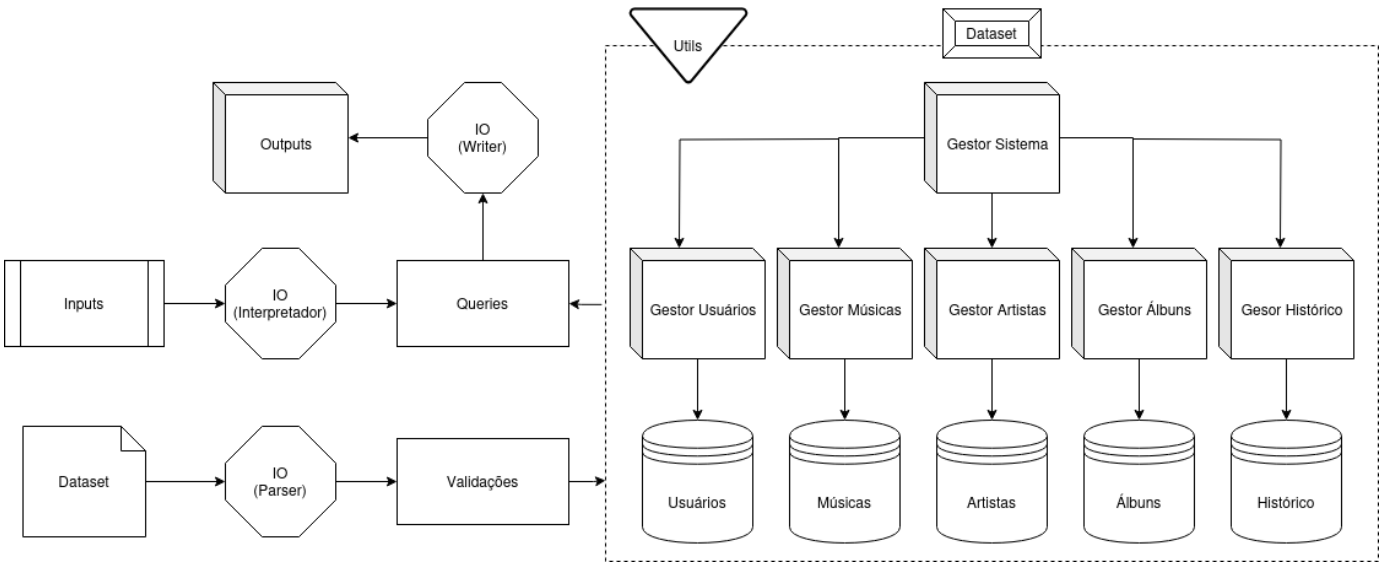
Finalizada a etapa inicial do projeto, que envolveu a implementação do parser, a validação dos dados e a execução de 3 das 6 queries, atingimos os objetivos estabelecidos para a segunda fase: conclusão das 3 queries restantes, ajuste de uma das já realizadas, desenvolvimento do modo interativo e realização de testes funcionais, para além da otimização do uso de memória e tempo de execução, necessários devido à maior dimensão do dataset.

2.Objetivos da 2ª Fase

Os objetivos da 2ª fase do projeto são:

- **Processar novos dados de musicas, álbuns e histórico** a partir de arquivos CSV .
- **Escrever os resultados das consultas** em arquivos de saída formatados de forma adequada, utilizando a funcionalidade de **writers**.
- **Gerir erros de execução e falhas** no processamento, filtrando informações inválidas e usando o novo programa de testes.
- **Implementação do modo interativo**.
- **Otimizações no desempenho**.

3. Arquitetura do Sistema



4.Ajustes da 2ªFase

4.1. Indicações dos Docentes

Foram-nos sugeridos alguns métodos de otimização pelos docentes durante a defesa da 1ªFase para aperfeiçoarmos o projeto:
- Aperfeiçoamento dos testes para dizer tempo por query e linha em que diferem;
- Pré-processamento.

4.2. Execuções das Queries

O sistema implementa um conjunto de consultas que permitem recuperar informações específicas a partir de dados armazenados. Cada consulta é invocada conforme os comandos presentes no arquivo *inputs.txt*, com o objetivo de processar rapidamente grandes volumes de dados.
Todas as queries, em termos de formatação, diferem num pequeno detalhe pois são representadas, nos inputs, por um número, seguido ou não do caractere 'S'. Os outputs deverão ser separados por ';', enquanto que o formato alternativo ('S') deverá utilizar o separador '='.

A **Query 1** passou a ter uma nova alteração que permite agora também recuperar as informações de um artista com base o seu ID. Para isso, o sistema faz uso de uma *Hashtable* (*GHastable*), que armazena os dados dos utilizadores de forma eficiente. A chave de busca, que é o *ID*, mapeia diretamente para a posição na tabela, permitindo que a consulta seja realizada de forma rápida. Uma vez localizada a entrada correspondente, as informações do artista, como *nome*, tipo, país, número de álbuns individuais e a receita total de um artista são formatadas conforme o especificado e escritas no ficheiro de saída. Se o artista não for encontrado, o sistema gera um ficheiro de saída vazio. Devido à receita total de um artista ser um valor do tipo double, encontramos algumas dificuldades para obter o valor correto em todos os outputs.

A **Query 4** tem como objetivo saber qual o **artista que esteve na top 10 mais vezes**. Para isso, o sistema pré-processa os dados dos artistas, calculando e armazenando o top 10 de cada semana encontrada no histórico, bem como o top 10 geral. Quando a query é chamada, é efetuada uma busca, levando em consideração um possível filtro de datas, nestes dados. Caso o filtro de datas seja fornecido, apenas os artistas ouvidos nesse intervalo de datas são considerados na consulta.

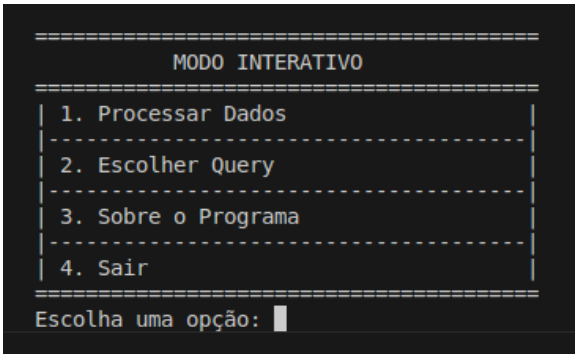
A **Query 5** tem como objetivo a recomendação de utilizadores com gostos parecidos. Para tal, em pré-processamento é criada uma lista com todos os utilizadores e uma lista com todos os géneros, e construída uma matriz em que cada linha corresponde a um utilizador e cada coluna corresponde a um género de música. Após ser chamada a query, é utilizado o recomendador e dado o output.

Por fim, a **Query 6** recebe dois argumentos: o id do utilizador para o qual queremos obter um resumo de estatísticas anuais e o ano ao qual essas estatísticas se referem. Para isso, o sistema processa os dados dos utilizador, levando em consideração um possível filtro a considerar, um argumento opcional numérico, N, sendo que em tal caso devem-se imprimir em linhas consecutivas os N artistas mais ouvidos pelo utilizador nesse ano, em função do tempo de reprodução.
A query 6 irá criar uma HashTable com todas as informações referentes ao ano que recebeu e por fim irá dar como output o resumo anual.

5. Novos Programas

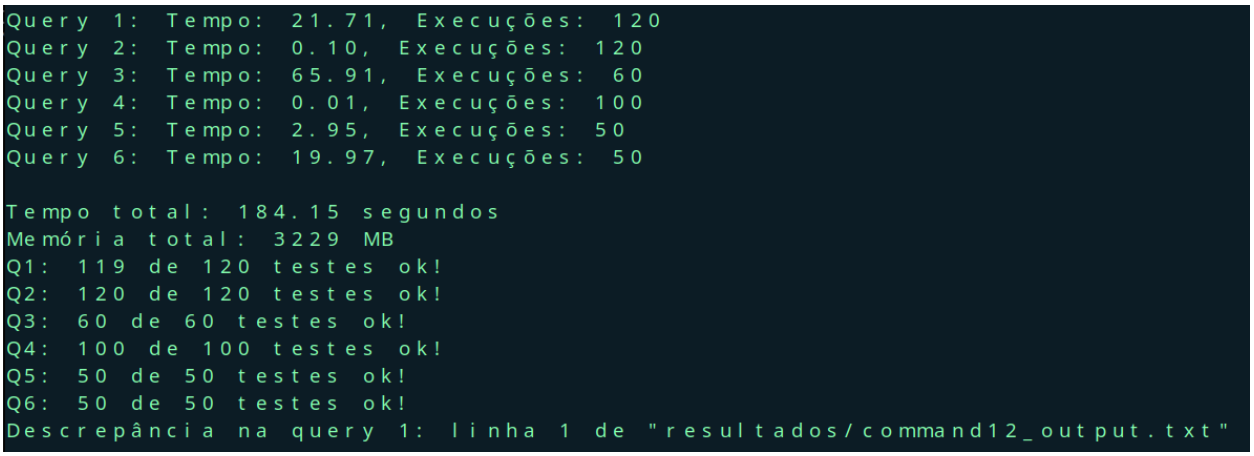
5.1. Progama Interativo

O programa interativo, um dos critérios da 2ª Fase, possibilita que o usuário informe ao programa o caminho dos arquivos que contêm os dados e execute as queries inseridas por ele.



5.2. Programa de Testes

O programa de testes valida a correta implementação das queries, bem como o desempenho do programa. As medidas observadas por esta ferramenta são o tempo que cada query leva para ser executada, bem como o tempo total e o uso de memória total, finalmente, indica onde há diferenças nos resultados.



6. Otimizações

De modo a otimizar o gasto de memória e o tempo de execução do programa, como o novo dataset de maiores dimensões, foi necessário aplicar algumas alterações ao programa. Essas foram:

- A introdução de pré-processamento de vários dados como o total de "streams" de cada música, a duração da discografia dos artistas, o número de artistas e géneros e o top 10 artistas de cada semana encontrada no histórico;
- A conversão dos IDs e Usernames do tipo string para o tipo inteiro;
- A não inserção de dados não relevantes para as queries, nomeadamente as lyrics de uma música, produtores de um álbum e descrição de um artista.

7. Conclusões

Concluindo, o projeto serviu como uma valiosa experiência no desenvolvimento de sistemas robustos e eficientes, proporcionando aprendizagens significativas sobre modularidade, encapsulamento, gestão de memória e a importância de testar e validar adequadamente cada componente do sistema. Considerámos ainda que foram atingidos pelo grupo os objetivos do projeto e demonstradas as aprendizagens da cadeira, o que foi possibilitado pela base sólida vinda da 1ª fase, já com a modularidade e o encapsulamento bem implementados.