

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Laboratórios de Informática

III

Trabalho prático (Fase 2)

Janeiro 2025

Desenvolvido por:

Rodrigo Ferreira - a104531
Gonçalo Alves - a104079
Diogo Esteves - a104004

Contents

1 Introdução

2 Sistema

- 2.1 Arquitetura
- 2.2 Novos Componentes
- 2.3 Estruturas de Dados e Modularização
- 2.4 Documentação do Código

3 Discussão

- 3.1 Parsing Genérico
- 3.2 Novas Queries
 - 3.2.1 Query 5: Recomendador de utilizadores
 - 3.2.2 Query 6: Resumo anual de um utilizador
- 3.3 Modo Interativo

4 Conclusão

1 Introdução

Este relatório tem como finalidade detalhar a evolução do projeto prático de Laboratórios de Informática 3 na Fase 2, que introduz novas funcionalidades e otimizações em um sistema de *streaming* de música. Esta fase expande as competências de modularização, encapsulamento e manipulação de grandes conjuntos de dados, com foco na documentação do código, no desempenho e na qualidade geral da solução.

O trabalho desenvolvido nesta fase incorpora novos dados (*albums* e *history*), novos modos de execução (*programa-interativo*), três novas queries (Q5-Q6) e melhorias significativas no processo de *parsing*. Além disso, são introduzidos critérios adicionais de avaliação, como a interatividade, otimizações de estruturas de dados e documentação padronizada com *Doxxygen*.

2 Sistema

2.1 Arquitetura

A estrutura do sistema foi ampliada para suportar os novos componentes introduzidos na Fase 2. Segue abaixo o diagrama atualizado da arquitetura da aplicação:

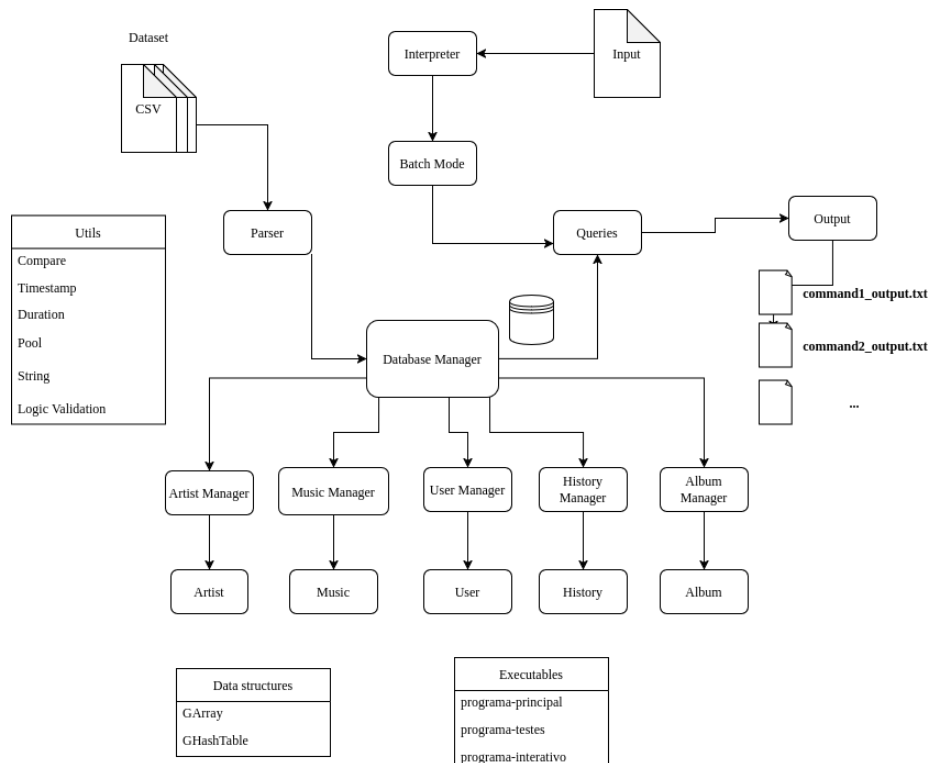


Figure 1: Arquitetura atualizada da aplicação

2.2 Novos Componentes

Nesta fase, os seguintes componentes foram adicionados ou expandidos:

- **Módulo de Histórico:** Permite armazenar e consultar eventos de reprodução, como tempo de audição e plataforma utilizada.
- **Módulo de Álbuns:** Garante o gerenciamento de álbuns, associando-os às suas respectivas músicas e artistas.

- **Modo Interativo:** Proporciona uma interface para execução dinâmica de queries via terminal.
- **Parsing Genérico:** Implementação de funções de parsing mais flexíveis e reutilizáveis, facilitando o carregamento de dados variados.

2.3 Estruturas de Dados e Modularização

As estruturas de dados foram otimizadas para lidar com o aumento do volume de informações. As *Hash Tables* continuam sendo a principal escolha para buscas rápidas, complementadas por *arrays dinâmicos* para listas de eventos. Para cada nova entidade, foram criados *managers* responsáveis pela manipulação segura e encapsulada dos dados.

2.4 Documentação do Código

Foi adotado o *Doxygen* para garantir documentação padronizada das funções e estruturas de dados, detalhando:

- **@brief:** Resumo da funcionalidade.
- **@param:** Descrição dos parâmetros.
- **@return:** Valor de retorno.

3 Discussão

3.1 Parsing Genérico

Uma das melhorias mais significativas na Fase 2 foi a substituição do parser específico por um modelo genérico. Este parser utiliza funções comuns para processar campos de diferentes entidades, reduzindo redundâncias e facilitando a manutenção do código. Por exemplo, a função de parsing genérica permite lidar com campos de *CSV* de forma mais flexível, conforme demonstrado abaixo:

```
while ((token = strsep(&aux, ";")) != NULL) {
    int len = strlen(token);
    if (len > 2) {
        token++;
        token[len - 2] = '\\0';
    }
    char *element = strdup(token);
```

Esta abordagem simplificou o carregamento de dados, melhorando tanto o desempenho quanto a legibilidade do código.

3.2 Novas Queries

3.2.1 Query 5: Recomendador de utilizadores

A Query 5 identifica utilizadores com gostos musicais similares ao de um utilizador alvo. Para isso, construímos uma matriz onde cada linha representa um utilizador e cada coluna, um género musical, com valores indicando o número de audições por género. A matriz foi preenchida iterando sobre a *HashTable* de utilizadores, utilizando o campo `user_history` de cada `user_t` para contabilizar os géneros das músicas reproduzidas.

Com os dados preparados, utilizamos a função `recomendaUtilizadores` da biblioteca fornecida, passando:

- **Matriz:** Frequência de audições por género para cada utilizador.
- **IDs dos utilizadores:** Array de IDs correspondente às linhas da matriz.
- **Nomes dos géneros:** Array fixo com os géneros musicais.
- **Parametro de recomendações:** Número de utilizadores semelhantes a retornar.

A adoção do campo `user_history` eliminou a necessidade de uma *HashTable* separada para eventos, simplificando o acesso aos dados. A matriz foi alocada dinamicamente com `calloc`.

3.2.2 Query 6: Resumo anual de um utilizador

Gera estatísticas detalhadas de audições de um utilizador, incluindo géneros mais ouvidos, álbuns favoritos e artistas preferidos. Esta query recebe como parametro opcional um número que, caso seja fornecido, lista os top N artistas mais ouvidos pelo utilizador.

3.3 Modo Interativo

O modo interativo introduzido nesta fase permite:

- Escolha dinâmica do *dataset* a ser carregado.
- Execução de queries com entrada direta do usuário.
- Validação de dados em tempo real, garantindo a robustez do sistema.

4 Conclusão

A segunda fase do projeto expandiu significativamente a complexidade e as funcionalidades do sistema, consolidando conceitos de modularização e encapsulamento. Enfrentamos desafios relacionados ao aumento do dataset e integração de novos módulos, mas conseguimos implementar soluções eficientes.

Para o futuro, planeamos explorar mais otimizações em queries complexas e integrar novas funcionalidades que aumentem o valor prático do sistema.