Introdução

Este trabalho prático, desenvolvido no âmbito da cadeira de Laboratórios de Informática III, tem como objetivo principal a construção de um programa estruturado (em vários módulos) em C para análise de dados de um sistema de streaming de música. O projeto tem foco na consolidação de algumas habilidades de programação em C - sobretudo nas boas práticas de modularização, mas também no encapsulamento e gestão da memória. O sistema, ao explorar uma grande quantidade de dados, como informações sobre músicas, artistas e users, proporciona uma experiência prática na manipulação e consulta de grandes volumes de dados.

A implementação do projeto está dividida em duas fases, sendo este relatório sobre a primeira. Esta fase foca no desenvolvimento e validação de funcionalidades básicas, incluindo o processamento e validação de dados a partir de arquivos CSV e a implementação de um conjunto de 3 queries. Cada query permite ao sistema responder a perguntas específicas sobre os dados processados, como a listagem de informações de users, identificação dos artistas com maior discografia e popularidade de géneros musicais em diferentes faixas etárias. Todas estas funcionalidades foram projetadas de forma modular .

A organização do projeto baseou-se em princípios de modularidade. Dividimos o sistema em módulos para tratamento de dados (parsing), queries e operações sobre estruturas de dados. Esta abordagem permite encapsular as operações de leitura e manipulação de dados em um único ficheiro, facilitando a reutilização de código e simplificando o processo de desenvolvimento e manutenção. Além disso, a estrutura modular possibilita que o programa seja executado em dois modos distintos (nesta fase): o modo principal, para armazenamento de dados e execução geral de queries e o modo de testes, que inclui validações automatizadas de cada query.

Para garantir a qualidade e a estabilidade do programa, demos especial atenção à gestão de memória e à prevenção de leaks, utilizando o gdb e o valgrind para nos ajudar a perceber onde poderiam estar os problemas relacionados à memória. Também demos atenção à validação dos dados: garantimos que os dados estavam de acordo com os formatos definidos (datas, e-mails, listas e tipos de subscrição). Dados inválidos foram colocados em arquivos de erro, facilitando a perceção de entradas incorretas.

A realização desta fase do projeto foi uma grande oportunidade para melhorar técnicas de programação em C, especialmente a organização modular mas também a boa gestão de memória. Os desafios enfrentados ao longo do desenvolvimento desta fase, sobretudo aqueles relacionados à eficiência das estruturas de dados, à qualidade do código e à eliminação dos memory leaks, certamente contribuíram para o nosso desenvolvimento como programadores.

Sistema

A arquitetura do sistema foi projetada com o objetivo de garantir modularidade, separando as tarefas em módulos diferentes que encapsulam cada etapa do processamento e consulta de dados. Este design modular não só facilita a manutenção e o aperfeiçoamento do código, mas também permite que futuras extensões e otimizações sejam incorporadas mais facilmente - de forma organizada e isolada.

1. Arquitetura Geral

O sistema é composto por um programa principal que realiza o processamento de dados e execução de queries, e um programa testes que automatiza a verificação da correção dos outputs. Abaixo estão descritos os módulos implementados e as suas respetivas responsabilidades, seguidos de um diagrama que mostra esta organização.

Programa Principal

1. **Inputs**: O sistema recebe dois arquivos principais como entrada: o Dataset, contendo os dados a serem processados (users, músicas, artistas), e o input.txt, que especifica as queries a serem executadas. Estes inputs são o ponto de partida para o fluxo de dados no sistema.
2. **Validação**: A primeira etapa do processo envolve o módulo de validação, que verifica a integridade dos dados do dataset. Dados incorretos ou inválidos são identificados e exportados para um arquivo errors.csv.
3. **Gestor de Dados**: Após a validação, os dados são organizados em três gestores específicos: **Gestor Artists**, **Gestor Musics** e **Gestor Users**. Cada um destes gestores encapsula as operações e dados relevantes ao seu domínio, mantendo as responsabilidades bem definidas. Para centralizar as interações entre os gestores é utilizado um **Gestor de Gestores.**
4. **Interpretação e Execução de Queries**: Depois de identificadas as queries a serem executadas é chamado o **Gestor Queries**. Este módulo coordena a execução das três queries principais do sistema, cada uma focada em responder perguntas específicas sobre o conjunto de dados:
    - **Query 1**: Listagem de informações específicas de users.
    - **Query 2**: Identificação dos artistas com maior discografia.
    - **Query 3**: Análise da popularidade de géneros musicais em diferentes faixas etárias.

    As respostas das queries são armazenadas nos ficheiros output1.txt, output2.txt, etc., dependendo do que foi especificado no input.txt.

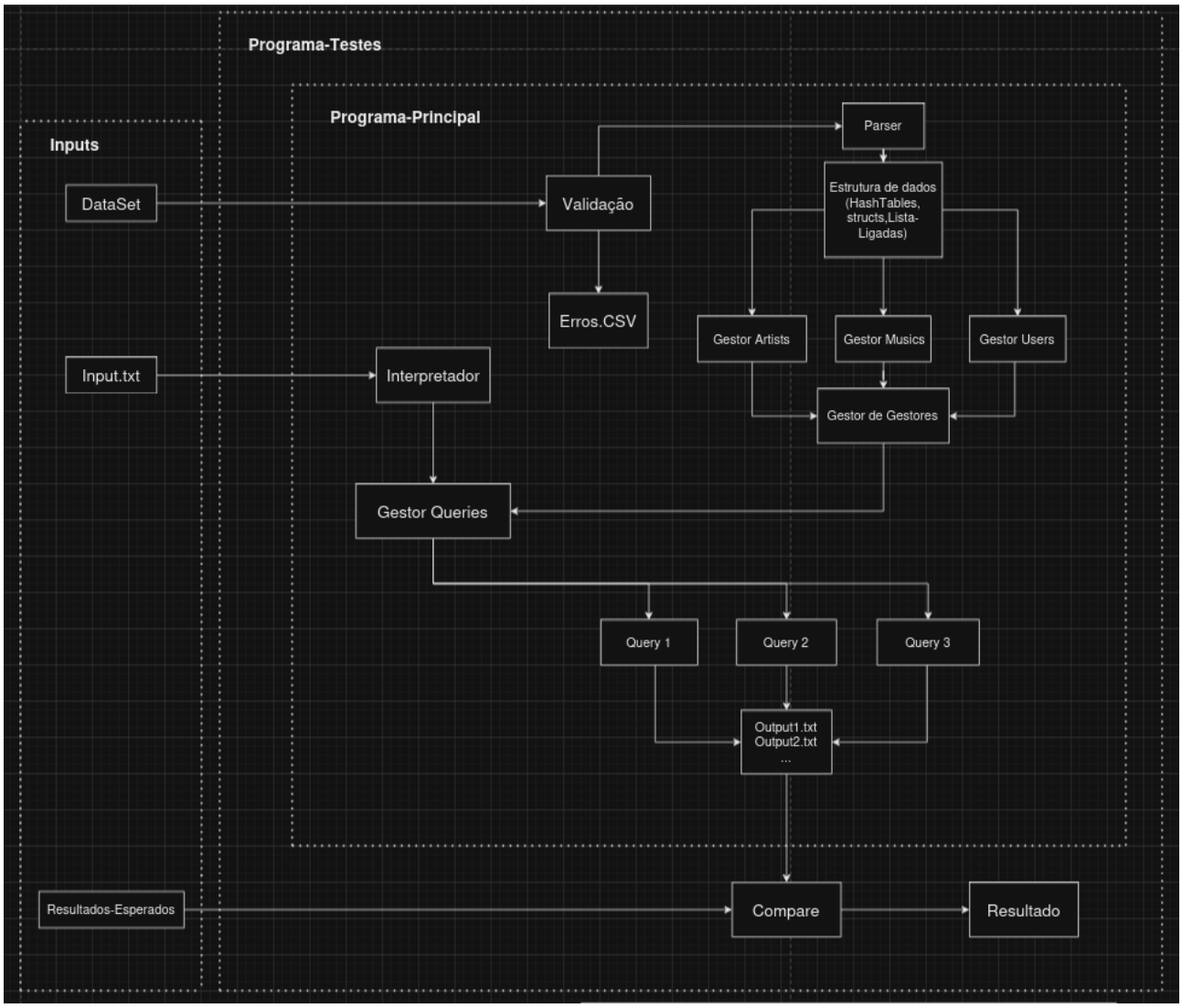
Programa Testes

O programa de testes foi desenvolvido para automatizar a verificação da correção do sistema. Ele compara os outputs gerados (output1.txt, output2.txt, ...) com os resultados esperados fornecidos, garantindo que o sistema responde corretamente às queries. Esse processo de comparação gera um resultado final, que indica o sucesso ou falha dos testes.

3. Justificativas das Escolhas de Modularidade

4. Diagrama da Arquitetura

Abaixo está o diagrama que representa a organização dos módulos:



Discussão

.

Conclusão

Nesta primeira fase do projeto desenvolvemos um programa modular em C para análise de dados de um sistema de streaming de música, aplicando técnicas fundamentais de programação e manipulação de dados. Através deste trabalho, consolidamos práticas de modularização e validação de dados, que foram essenciais para a implementação de um sistema capaz de processar e consultar grandes volumes de informações.

A principal lição aprendida foi a importância de uma organização modular e da separação de tarefas em diferentes componentes do sistema. Esta abordagem tornou o desenvolvimento mais eficiente e facilitou a manutenção e testes de cada módulo de forma independente, além de facilitar futuras extensões.

A experiência/utilização de ferramentas como o gdb e o valgrind para a gestão de memória foi também muito importante, pois permitiu/facilitou a identificação e resolução de leaks de memória. Este processo ajudou-nos a consolidar os conceitos de alocação e libertação de memória, algo essencial em C, especialmente ao lidar com grandes volumes de dados.

Achamos que o nosso sistema tenha apresentado um bom desempenho nesta Fase 1.

Em resumo, esta fase proporcionou-nos uma experiência prática na programação modular em C e também na manipulação de dados de grande dimensão. Depois de todos os desafios que enfrentamos na realização desta fase do projeto acreditamos que crescemos como programadores!