

Projeto Laboratórios de Informática III

Fase II

Projeto desenvolvido por:

Ricardo Gomes de Sousa (A104524),

Tomás Afonso Brito Oliveira (A104345),

Pedro Miguel de Abreu Argainha (A104351)

Grupo 62

Licenciatura em Engenharia Informática



Universidade do Minho
Escola de Engenharia

Departamento de Informática
Universidade do Minho

Índice

Introdução	3
Desenvolvimento.....	4
Modo Interativo	6
Dificuldades sentidas	10
Conclusão.....	11

Capítulo 1 - Introdução

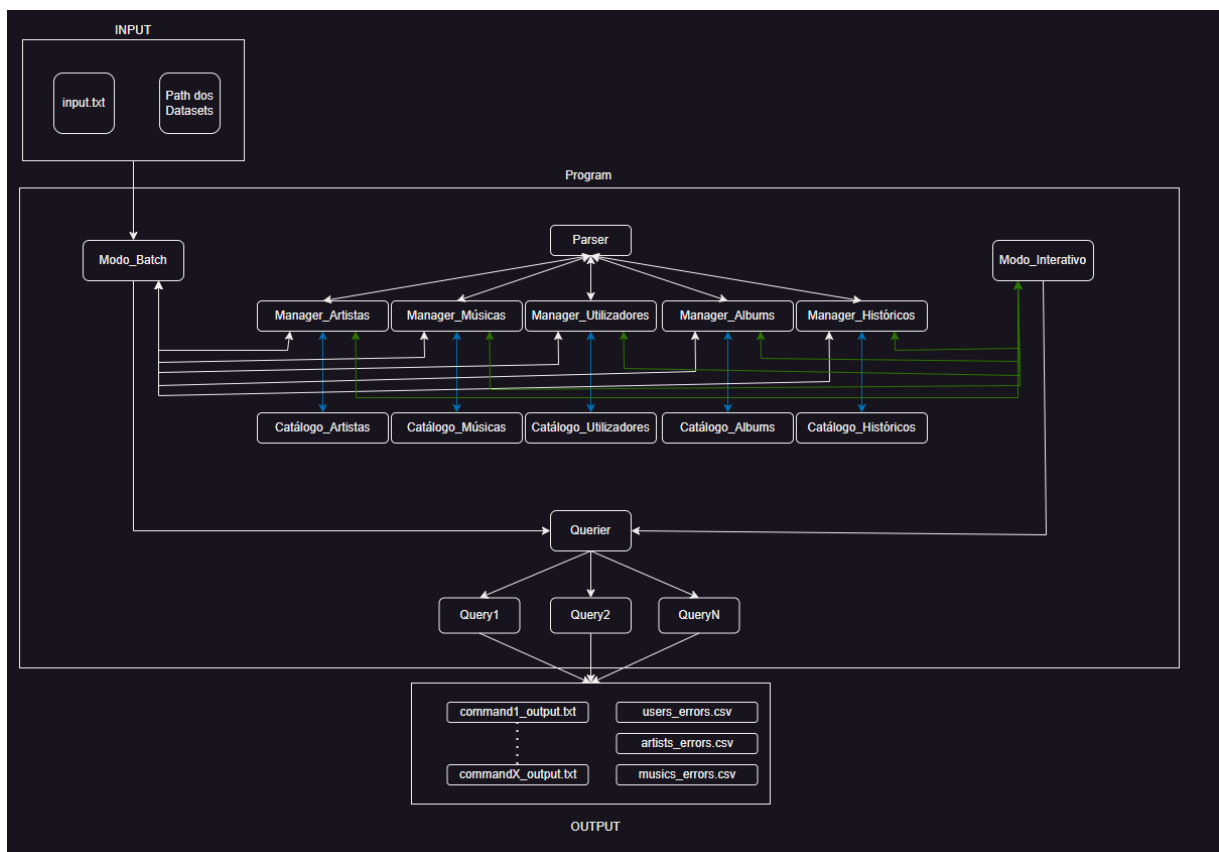
No âmbito da unidade curricular Laboratórios de Informática III (2024/2025), foi-nos proposto o desenvolvimento de um trabalho prático que envolve a implementação de uma base de dados em memória para manipular dados fornecidos pelos docentes. O objetivo deste projeto é consolidar conhecimentos essenciais na linguagem C, com foco nos seguintes aspetos: modularidade e encapsulamento, estruturas dinâmicas de dados, validação funcional, e avaliação de desempenho (computacional, consumo de memória, etc.). Pretende-se também adquirir experiência no uso de ferramentas essenciais ao desenvolvimento em C, como debugadores, ferramentas de análise de memória (e.g., Valgrind), compilação, linkagem.

O trabalho está dividido em duas fases. A Fase 2 introduz o modo de execução interativo, onde o programa é executado sem argumentos e disponibiliza um menu para interagir com os dados carregados. Neste modo, o utilizador pode introduzir o caminho do dataset a processar e executar queries de forma individual. Cada comando é processado e o respetivo resultado apresentado no terminal, com validação dos dados introduzidos para evitar comportamentos erráticos.

Esta fase, além de avaliar a funcionalidade interativa, promove o uso de técnicas avançadas de modularidade, encapsulamento e otimização do desempenho e recursos, refletindo os objetivos de aprendizagem da unidade curricular.

Capítulo 2 - Desenvolvimento

Para facilitar a execução e organização do código, o nosso grupo optou por começar com uma fase de planeamento estruturado. Desta forma, obtivemos uma visão mais clara do que já temos e do que pretendemos desenvolver, minimizando a probabilidade de termos de reestruturar partes significativas do código devido a implementações pouco eficientes ou práticas. Iniciámos este processo ao idealizar uma arquitetura que descrevesse a interação entre os diferentes módulos, bem como a forma como estes comunicariam entre si. Com base nesse planeamento, definimos a seguinte arquitetura:



QUERIES

Query 1:

A query 1 consiste em listar o resumo de um utilizador ou artista, consoante o identificador recebido por argumento. Dado o identificador, se este for relativo a um utilizador, é de seguida recolhida informações específicas sobre o mesmo, sendo a data de nascimento o único dado que necessita de ser processado para obter a idade do utilizador através da função *calcular_idade*. Se o identificador for relativo a um artista, é recolhida informações específicas sobre o mesmo. Aqui, o id do artista é usado para verificar em quantos álbuns este surge como autor. Dependendo da flag da query ("1" ou "1S"), os dados são apresentados de acordo com a estrutura dada no enunciado.

Query 2:

No início do programa a função *top_artistas* itera pela hashtable de músicas. Em cada música passa pelo id dos autores e procura-os na hashtable de artistas. A cada um soma a duração da música a variável duração dentro da entidade, e guarda um apontador para esse artista dentro de uma nova estrutura de dados, uma lista ligada. No final da iteração, organiza os artistas de forma decrescente. Quando a query é chamada vai a essa estrutura encontrar os primeiros N artistas. Caso seja pedido um país em específico, encontra os primeiros N artistas com esse país. Por fim devolve uma string com as informações necessárias desses artistas.

Query 3:

A query 3 consiste em listas quais os géneros de música mais populares numa determinada faixa etária. Dada a faixa etária, é de seguida percorrida a hashtable de utilizadores e caso este esteja dentro da faixa etária, verifica-se a sua lista de gostos. É verificada qual o género de cada música e incrementa-se o contador do género correspondente. No final teremos cada contador com o número de gostos de todos os utilizadores da faixa etária. De seguida, é escrito no ficheiro de output esses resultados de forma decrescente. Dependendo da flag da query ("3" ou "3S"), os dados são apresentados de acordo com a estrutura dada no enunciado.

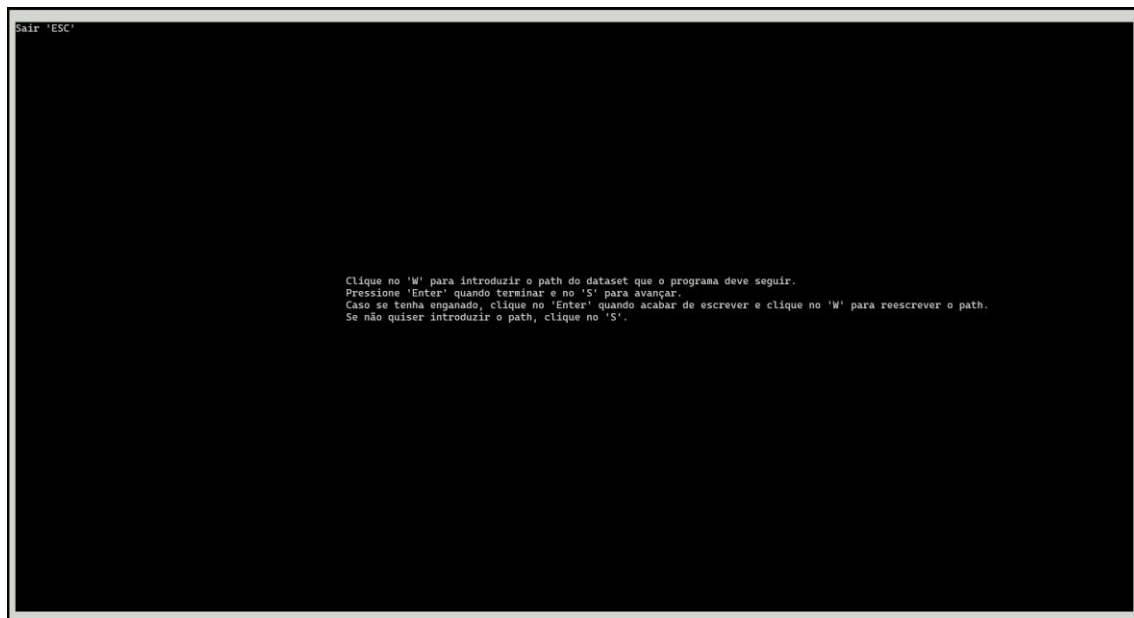
Capítulo 3 - Modo interativo

Inicialização:

O programa começa por inicializar a interface do menu utilizando a função `initscr()` da biblioteca *ncurses*. Em seguida, configura várias opções, como a ativação do teclado para entrada de dados e desativação da ecoação de caracteres, utilizando as funções `keypad()`, `noecho()` e `curs_set(0)`, que também ocultam o cursor.

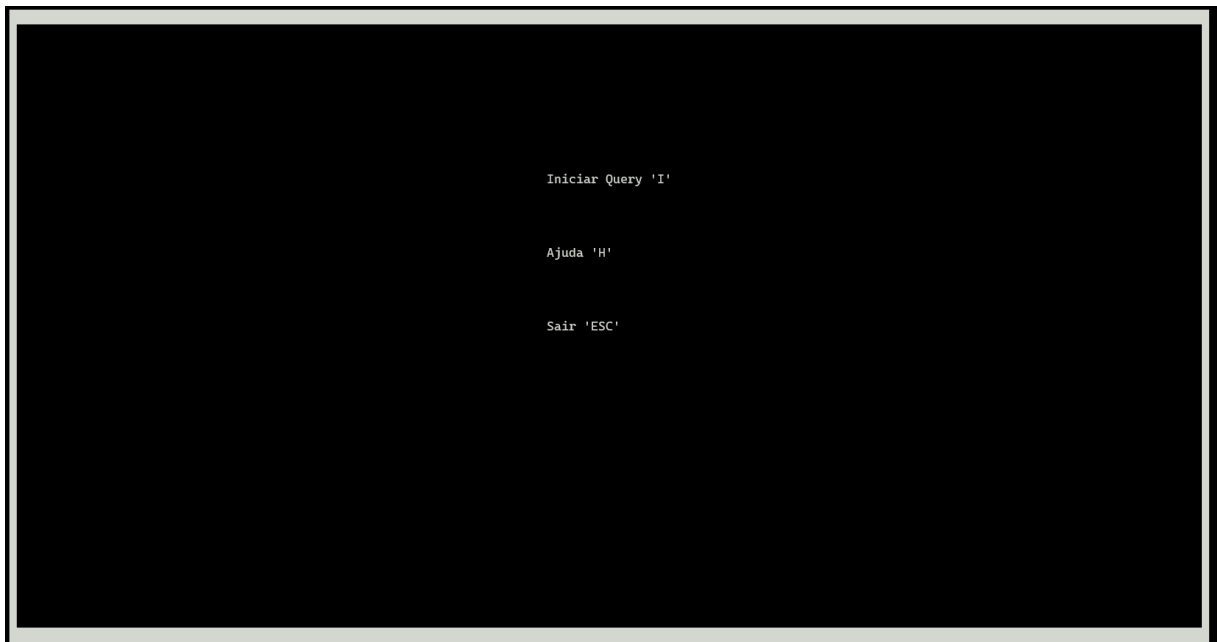
Menu Path:

O programa começa no menu path onde pede ao utilizador o path dos datasets a introduzir. Para escrever, começa por clicar no 'W' e depois escreve o path. Quando terminar, pressiona 'Enter' para confirmar o que escreveu e no 'S' para avançar. Caso não queira introduzir um path, clica apenas no 'S'.



Menu principal:

Depois de os datasets serem carregados e processados, o utilizador pode seleccionar entre 3 opções no menu principal: iniciar uma query, obter ajuda e sair.



Menu ajuda:

Ao seleccionar a opção de ajuda, vai para o menu de ajuda onde pode escolher qual a query que quer saber mais:



```
Voltar 'V'
```

Q1 - Listar o resumo de um utilizador ou artista, consoante o identificador recebido por argumento.

A query recebe como argumento o identificador único de um utilizador ou artista, sendo garantido que não existem identificadores repetidos entre as diferentes entidades. Deverá ser retornada uma linha vazia caso o id não conste do sistema. Nos casos em que conste, a query deverá retornar as seguintes informações:

- Utilizador
email;first_name,last_name;age;country
- Artista
name;type;country;num_albums_individual;total_recipe

Menu query:

Se o utilizador escolher a opção de iniciar uma query, é lhe de seguida pedido que introduza o comando que o programa deve correr, no formato especificado no enunciado:

```
Voltar 'V'
```

Introduza o comando a executar respeitando as seguintes instruções:

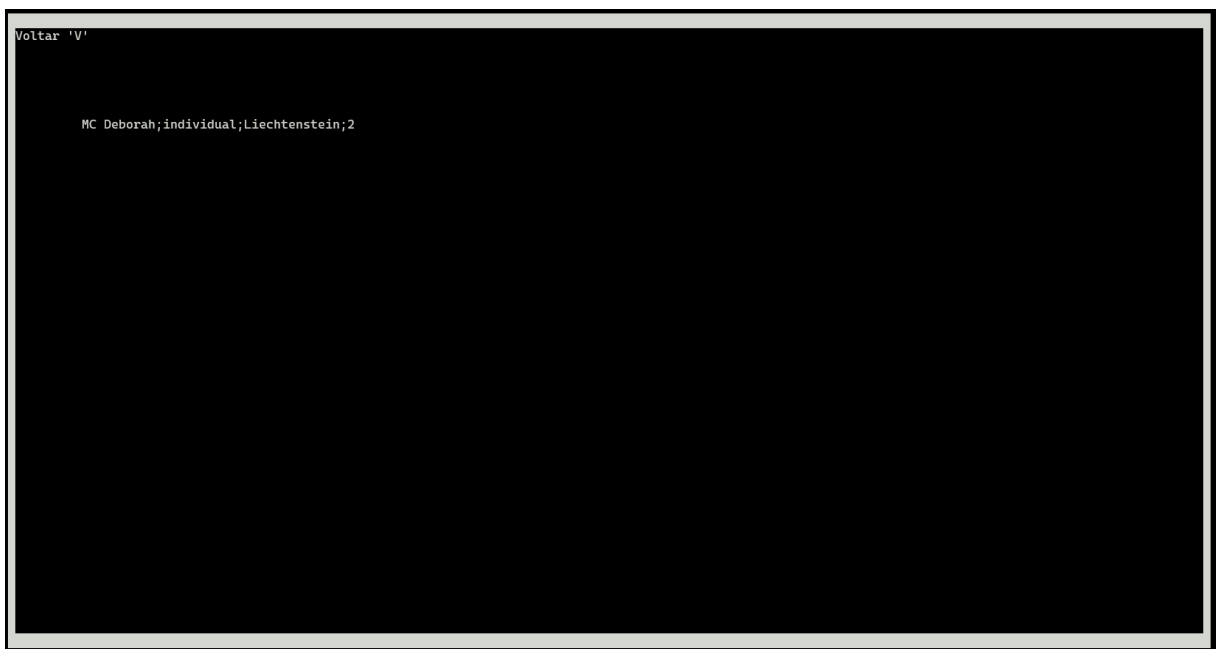
- o formato do comando deve ser "<query-id>[format-flag] [arg1...argN]"

Exemplos: "1 <ID> "; "15 <ID>"

Clique no 'W' para introduzir a query. Pressione 'Enter' quando terminar e no 'S' para avançar.
Caso se tenha enganado, clique no 'Enter' quando acabar de escrever e clique no 'W' para reescrever a query.

Menu output:

Após realizar a query pedida, o programa apresenta o resultado do comando no ecrã:



Sair do Programa:

Em qualquer momento, o utilizador pode optar por sair do programa seleccionando a opção "SAIR" no menu principal.

Capítulo 4 – Dificuldades Sentidas

Durante o desenvolvimento desta fase do projeto, enfrentámos várias dificuldades que impactaram significativamente o progresso. Não conseguimos realizar as *queries* 4, 5 e 6 devido a uma falta de compreensão sobre a sua resolução, o que limitou o avanço nesta área específica. Além disso, o início tardio do trabalho nesta fase dificultou a gestão do tempo disponível. A ausência de uma comunicação eficaz entre os membros do grupo agravou o problema, resultando em atrasos na execução e conclusão das tarefas previstas para esta etapa.

Capítulo 5

-

Conclusão

Em resumo, este projeto representou ser um grande desafio para o nosso grupo, especialmente nesta segunda fase. Isso deve-se à introdução de vários conceitos novos e de implementação complexa, que exigiram um esforço adicional. Apesar dos resultados desta fase não terem sido tão positivos quanto os da anterior, sentimos que conseguimos melhorar a qualidade do nosso trabalho em alguns aspetos.

Reconhecemos que, enquanto grupo, poderíamos ter colaborado de forma mais eficaz. Em projetos futuros, pretendemos continuar a evoluir e a ultrapassar as fragilidades identificadas.