

Projeto Laboratórios de Informática III

Fase 1

Projeto desenvolvido por
Fábio Leite (A100902), Luís Figueiredo (A100549) e Gonçalo
Gonçalves (A100833)

Grupo 50

Licenciatura em Engenharia Informática



Universidade do Minho
Escola de Engenharia

Departamento de Informática
Universidade do Minho

Conteúdo

1	Introdução	2
2	Desenvolvimento	3
3	Dificuldades sentidas	6
4	Conclusão	7

Capítulo 1

Introdução

Este projeto está a ser desenvolvido no âmbito da unidade curricular de Laboratórios de Informática III do ano 2023/2024. Neste projeto é nos instruída a implementação de uma base de dados em memória que armazene dados fornecidos em ficheiros ".csv".

O projeto está dividido em duas fases, e esta primeira fase consiste em implementar o *parsing* dos dados e em executar várias *queries* sobre os dados, estando esses pedidos armazenados num ficheiro de texto cujo caminho é recebido como argumento.

Para a execução deste projeto foi usada a biblioteca *glib* uma vez que esta biblioteca possui implementações de várias estruturas úteis.

Capítulo 2

Desenvolvimento

No desenvolvimento da aplicação começamos por imaginar uma arquitetura que representaria a ligação entre os diferentes ficheiros que iriam compor o programa e a forma como iriam comunicar entre si, tentando garantir a existência do encapsulamento e modularidade durante o desenvolvimento do projeto. Após alguma discussão sobre a estrutura que iríamos adotar, chegamos ao seguinte modelo:

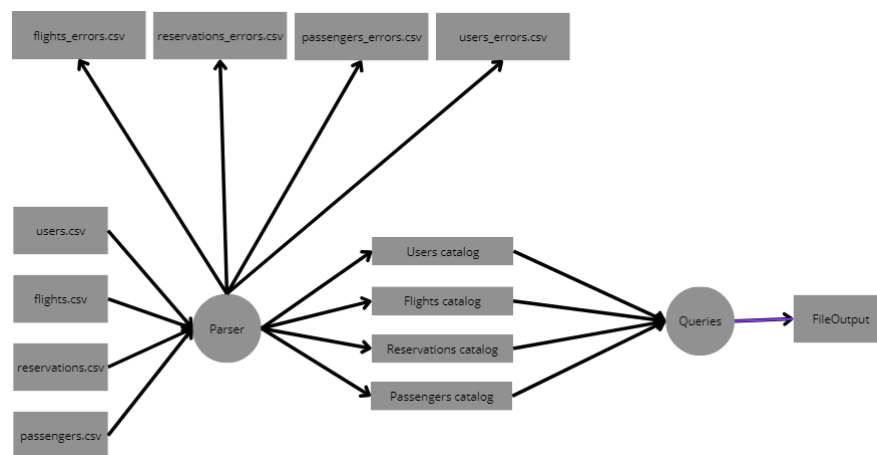


Figura 2.1: Diagrama da estrutura utilizada para o projeto

O passo inicial foi perceber como devíamos armazenar os dados. Optamos por usar *hashtables* para os ficheiros que apresentavam mais dados (*users.csv*, *flights.csv*, *reservations.csv*) devido ao facto de estas oferecerem um tempo de pesquisa constante. Reparámos que o *passengers.csv* apresenta apenas dois elementos: *user_id* e *flight_id*. Estes id's podem ser repetidos ao longo do ficheiro e, portanto, não faria sentido usar nenhum deles como *key* de uma *hashtable*. Foi decidido, então, guardar os dados deste ficheiro numa lista ligada (GList). Já na parte prática, (tendo sempre a

modularidade em conta) criámos *structs* para guardar cada linha do ficheiro, dividindo os diferentes campos da mesma nos respetivos atributos. Depois implementámos os catálogos de modo a armazenarem várias destas *structs* (PASSENGER, FLIGHT, USER e RESERVATION).

Críamos o *Parser* que é responsável por tratar os ficheiros .csv na pasta cujo caminho é fornecido como 1º argumento ao executar o programa e, posteriormente, encaminhar os dados para as respetivas estruturas onde a informação vai ser guardada. O *parser* chama funções do módulo *Validation* ao fazer o seu trabalho de modo a garantir a validação dos dados armazenados. Em caso de erro, envia-os para o módulo *Output* que escreve nos ficheiros de erros correspondentes todas as linhas do ficheiro original que continham qualquer erro.

Depois disto, implementámos o módulo *Interpreter* que é responsável por interpretar os comandos passados pelo ficheiro .txt de input passado como 2º argumento ao executar o programa, chamando a função correspondente implementada no módulo *Queries*, onde é permitido o acesso aos catálogos para que o resultado da *query* seja retornado (já devidamente formatado) ao *Interpreter*, que, por sua vez, o envia para o *Output* para que este possa escrevê-lo devidamente em ficheiros commandX_output.txt que, caso não existam, são criados.

Isto tudo tendo em conta as regras de encapsulamento que, apesar de adicionar uma camada extra de complexidade ao código por ser mais difícil acessar e modificar certos dados, permitiram-nos desenvolver um sistema mais robusto e modular ao evitar acesso não autorizado e facilitando a manutenção do código a longo prazo.

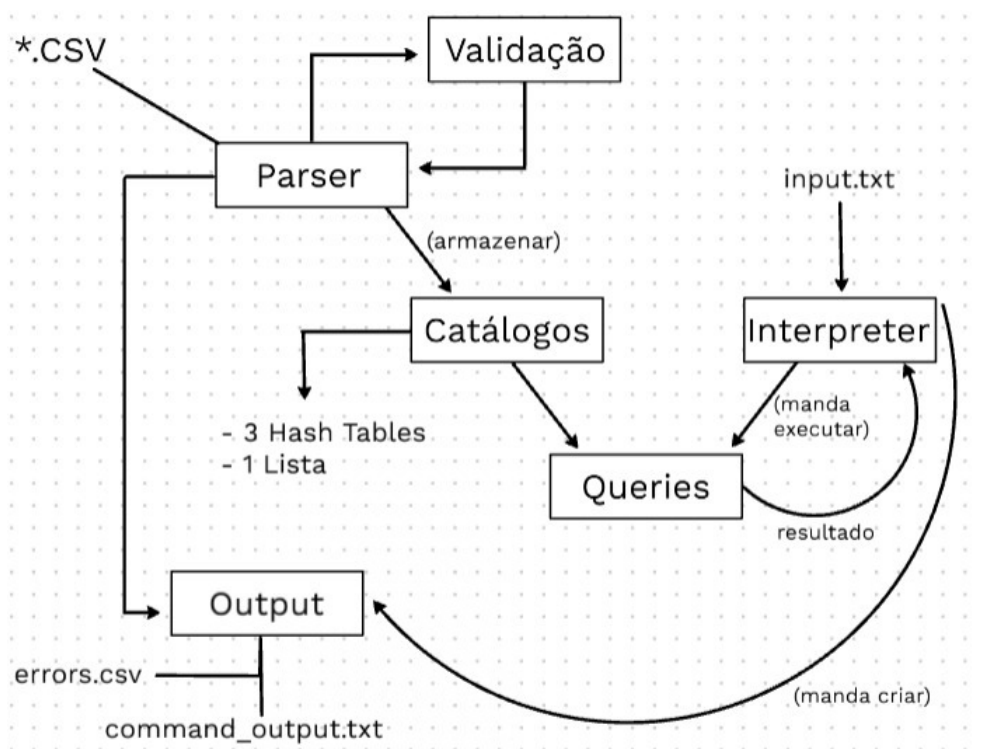


Figura 2.2: Diagrama do funcionamento do programa

Capítulo 3

Dificuldades sentidas

Ao longo do desenvolvimento do projeto, fomos encontrando diversos problemas que, para além de produzirem resultados errados, também impediram o nosso progresso. Primeiramente, sentimos uma grande dificuldade na validação de certos dados, sendo que alguns deles (na nossa perspetiva) só poderiam ser validados após a inserção dos mesmos nas nossas estruturas.

Da mesma forma, certas queries levantaram-nos dúvidas, especialmente a query 9. Nesta query, a nossa dúvida principal é provocada pelo facto de a mesma provocar *segmentation fault* na plataforma de testes, embora a mesma não o faça localmente.

Por fim, a utilização da *glib*, embora traga bastantes benefícios, também traz uma maior complexidade para o projeto, sendo necessários vários momentos de pesquisa e teste para perceber as funcionalidades que as estruturas de dados e as funções desta mesma biblioteca apresentam.

Capítulo 4

Conclusão

Dado o problema proposto, no nosso ponto de vista, a resolução por nós implementada cumpre grande parte dos principais objetivos desta primeira fase. Para o armazenamento da informação recolhida pelo parsing dos dados dos ficheiros .csv, foi por nós usado hash tables por considerarmos o método mais eficiente e rápido na procura dos dados para a resolução das queries propostas e uma lista ligada uma vez que a utilização de uma tabela de hash neste caso não seria justificada.

Embora apresentemos 6 queries no nosso projeto, algumas delas não se apresentam corretamente implementadas, sendo esse um dos principais pontos a melhorar para a próxima fase. Da mesma forma, também pretendemos melhorar a qualidade do código para próxima fase, uma vez que o consideramos um pouco confuso.