

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Sistema de Gestão de Vendas
Grupo N.º 1

Maria Silva (A83840)

Miguel Solino (A86435)

Paulo Lima (A89983)

12 de Abril de 2020

Conteúdo

1	Introdução	3
2	Problema	4
3	Módulos e API	6
3.1	Cliente	6
3.2	Catálogo de Clientes	6
3.3	Produto	6
3.4	Catálogo de Produtos	6
3.5	Venda	6
3.6	Fatura	6
3.7	Catálogo de Faturas	7
3.8	Filial	7
3.9	Catálogo de Filiais	7
3.10	SGV	7
3.11	Controller	7
3.12	View	7
3.13	Argv	7
4	Arquitetura do projeto	8
5	Estruturas de dados	9
6	Testes e Benchmarks	10
7	Conclusão	11
A	Grafo de Dependências	12
B	Tabela de Tempos de Execução	13

Capítulo 1

Introdução

O objetivo deste projeto é construir um sistema de gestão de vendas, sendo este capaz de armazenar informações de vendas realizadas relacionando com produtos e clientes existentes. A arquitetura principal aplicada neste projeto é MVC (model, view e controller) e foi necessário garantir encapsulamento dos dados armazenados de forma a serem só alterados através de agentes internos.

No decorrer do relatório, serão apresentadas as nossas abordagens na realização deste trabalho e no final é apresentado testes de performance do mesmo.

Capítulo 2

Problema

Foram nos fornecidos 3 ficheiros principais e 2 para testes:

- Ficheiro de produtos - contém códigos de produtos para inserir no catálogo de produtos
- Ficheiro de clientes - contém códigos de clientes para inserir no catálogo de clientes
- Ficheiro de vendas de 1 milhão - contém linhas de vendas com informações a serem tratadas
- Ficheiro de vendas de 3 milhões e 5 milhões

Usando estes ficheiros foi nos pedido para responder a 13 queries:

1. Ler os ficheiros fornecidos dando a opção de escolha se pretende usar os padrão ou outros inseridos pelo utilizador.
2. Lista e o n^o total de produtos cujo código se inicia por uma dada letra (maiúscula);
3. Dado um mês e um código de produto, ambos válidos, determinar e apresentar o número total de vendas (n^o de registos de venda) e o total facturado com esse produto em tal mês, distinguindo os totais em modo N e os totais em modo P. O utilizador deverá decidir se pretende o resultado global ou os resultados filial a filial para todas as 3 filiais.
4. Determinar a lista ordenada dos códigos dos produtos(e o seu número total) que ninguém comprou, podendo o utilizador decidir igualmente se pretende valores totais ou divididos pelas filiais.
5. Determinar a lista ordenada de códigos de clientes que realizaram compras em todas as filiais;
6. Determinar o número de clientes registados que não realizaram compras bem como o número de produtos que ninguém comprou.
7. Dado um código de cliente, criar uma tabela com o número total de produtos comprados (ou seja a soma das quantidades de todas as vendas do produto), mês a mês (para meses em que não comprou a entrada deverá ficar a 0). A tabela deverá ser apresentada em ecrã organizada por filial.
8. Dado um intervalo fechado de meses, por exemplo [1..3], determinar o total de vendas (n^o de registos de venda) registadas nesse intervalo e o total faturado;
9. Dado um código de produto e uma filial, determinar os códigos (e número total) dos clientes que o compraram, distinguindo entre compra N e compra P;
10. Dado um código de cliente e um mês, determinar a lista de códigos de produtos que mais comprou por quantidade e não por faturação), por ordem decrescente;

11. Criar uma lista dos N produtos mais vendidos em todo o ano, indicando o número total de clientes e o número de unidades vendidas, filial a filial;
12. Dado um código de cliente determinar quais os códigos dos N produtos em que mais gastou dinheiro durante o ano;
13. Apresentar resultados da leitura dos ficheiros da query 1. Apresentação imediata ao utilizador do nome do ficheiro lido e que foi usado, bem como o número total de linhas lidas e validadas

Capítulo 3

Módulos e API

3.1 Cliente

Neste módulo é onde são tratados os clientes individualmente. Aqui todos os códigos de clientes recebidos são tratados, e validados e é onde estão presentes as funções auxiliares para a resposta às queries relacionadas com clientes.

3.2 Catálogo de Clientes

Neste módulo é onde são adicionados, consultados e removidos os clientes ao catálogo. Todos os clientes aqui presentes são todos validados antes de serem inseridos. Como estrutura principal de armazenamento dos dados decidimos usar Hashtables da biblioteca Glib.

3.3 Produto

Semelhante ao módulo dos clientes, neste módulo é onde são tratados os produtos individualmente. Aqui todos os códigos de produtos recebidos são tratados, e validados e é onde estão presentes as funções auxiliares para a resposta às queries relacionadas com produtos.

3.4 Catálogo de Produtos

Neste módulo é onde são adicionados, consultados e removidos os produtos ao catálogo. Todos os produtos aqui presentes são todos validados antes de serem inseridos. Como estrutura principal de armazenamento dos dados decidimos usar Hashtables da biblioteca Glib.

3.5 Venda

Neste módulo são tratadas todas as linhas de vendas dos ficheiros com entradas de vendas de forma individual. Cada linha no momento de serem tratadas são divididas por uma estrutura que possui uma entrada para cada campo de informação da linha de venda. Desta forma, ao retornarmos os dados será mais fácil de se tratar.

3.6 Fatura

Neste módulo, depois da linha de venda ser tratada no módulo anterior, os dados são inseridos/atualizados nas estruturas aqui presentes. Essas estruturas contém informações relativas a faturação

global com valores totais relacionados com os produtos. Aqui também são usadas Hashtables de Glib.

3.7 Catálogo de Faturas

Este módulo contém o módulo anterior separado por 12 meses, ou seja, para cada mês existe uma fatura com os valores totais das vendas para cada produto e global. Como na anterior, as Hashtables da Glib foram a escolha.

3.8 Filial

Tal como no módulo da Fatura, neste contém o mesmo género de informações mas separado por relações. Assim aqui são guardadas as informações sobre as relações de produtos com clientes e de clientes e produtos. Como nas estruturas anteriores, aqui também foi usado Hashtables da Glib.

3.9 Catálogo de Filiais

Estando este módulo relacionado com o anterior, aqui são guardados as informações do anterior mas separado pelas filiais existentes, utilizando também Hashtables como tipo de estrutura principal.

3.10 SGV

O módulo SGV é a junção de todos os módulos e é onde estão presentes as funções principais chamadas para responder às queries. Aqui é onde o controlador faz os pedidos e é através deste que também inicializa as estruturas da base de dados, também estas utilizando Hashtables como principal estrutura.

3.11 Controller

Aqui neste módulo é onde é feita a comunicação entre o utilizador e o sistema. Através deste perguntamos ao utilizador que query pretende resolver. Este depois da resposta do utilizador contacta o model para receber os dados e mostra os utilizados a view.

3.12 View

O módulo View é onde estão presentes todas as funções genéricas para impressão dos dados como também cada função específica para mostrar os resultados das queries.

3.13 Argv

Este módulo permite realizar *benchmarks* a queries através dos argumentos da linha de comandos. Para tal várias funcionalidades foram implementadas:

1. `./SGV 1`: Cronometrar os tempos de carregamento com parâmetros default.
2. `./SGV 1 Path Path Path`: Cronometrar os tempos de carregamento com ficheiros definidos pelo utilizador.
3. `./SGV Query Parametros`: Cronometrar os tempos de resposta de cada query em específico.

Capítulo 4

Arquitetura do projeto

Este projeto segue uma Arquitetura do tipo *MVC* (Modelo, Apresentação e Controlador). Cada uma tem uma função específica e ajuda a separar a forma como está organizado o projeto. O model é o que está encarregue de inserir, tratar e remover os dados. O controller é o que utiliza o model para responder às queries e utiliza a view para mostrar os dados ao utilizador. Por último, a view é a responsável por imprimir os dados fornecidos pelo controller no ecrã de forma a que o utilizador consiga interpretar.

Capítulo 5

Estruturas de dados

Como descrito no capítulo 3, todas as nossas estruturas de dados consistem em Hashtables da GLib, tendo sido esta decisão tomada após testes realizados na primeira aula.

No início foi nos pedido para que utilizássemos arrays de forma a aprendermos como recolher dados de ficheiros de texto. No entanto, foi nos recomendado escolher uma das estruturas da Glib ou construirmos a nossa. Como no semestre passado tivemos a possibilidade de ter Algoritmos e Complexidade, já viemos com uma perceção de qual seria a melhor estrutura para usar neste tipo de projeto.

Capítulo 6

Testes e Benchmarks

Durante a execução do nosso projeto, foram efetuados diversos testes ao código, com ferramentas como *Valgrind*, para verificar a presença de *memory leaks*, *memusage*, para fazer profiling à memória utilizada, *gprof* para ver quais funções pesam mais na execução do programa, com vista a otimizar as partes mais lentas, e para medição dos tempos de execução utilizamos a biblioteca *time.h*.

No fase final do trabalho o nosso objetivo era corrigir todos os memory leaks e erros presentes no projeto. Para isso utilizamos a ferramenta *Valgrind*, utilizando a nossa funcionalidade de teste de tempos do programa, que nos indica todos os sitios onde existem memory leaks, quantidade de memória que não foi corretamente libertada.

Capítulo 7

Conclusão

Para concluir, tendo em conta os requisitos propostos, consideramos que conseguimos implementar tudo o que nos foi pedido de forma eficiente. Todas as queries são respondidas em tempos relativamente baixos, exceto uma que está acima de 1 segundo de resolução. Logo, como trabalho futuro gostaríamos de conseguir minimizar os tempos de carregamento de dados, diminuir o tempo de resolução da query 11 e manter constantes os tempos de resolução no geral.

Apêndice A

Grafo de Dependências

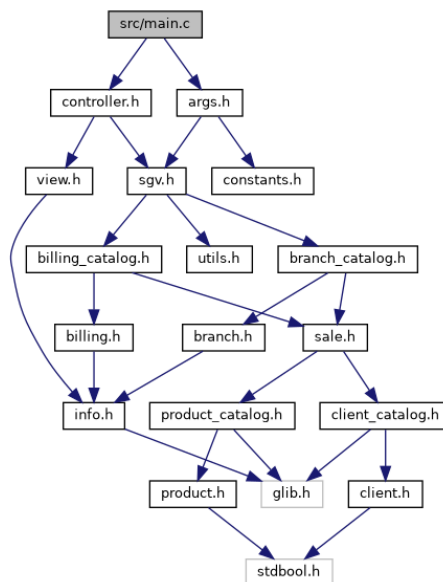


Figura A.1: Grafo simulado com todas as dependências

Apêndice B

Tabela de Tempos de Execução

	1 Milhão	3 Milhões	5 Milhões
Load Time	2.77	7.80	12.90
Query 2	0.005	0.005	0.005
Query 3	0.000002	0.000002	0.000002
Query 4	0.18	0.18	0.18
Query 5	0.009	0.009	0.009
Query 6	0.07	0.09	0.09
Query 7	0.00001	0.00006	0.00007
Query 8	0.000002	0.000002	0.000002
Query 9	0.000008	0.000008	0.000008
Query 10	0.00003	0.00006	0.00006
Query 11	0.40	1.13	1.80
Query 12	0.00001	0.00001	0.0002
Query 13	0.00000	0.00000	0.0000

Tabela B.1: Tempo (em segundos) das queries para um dado número de vendas