



Universidade do Minho

Relatório LI3

Sistema de Gestão de Vendas

Laboratórios de Informática III

MIEI - 2º ano - 2º semestre
2019/2020

Projeto em C

Realizado pelo Grupo 10 :
Rúben Correia Cerqueira (A89593)
Júlio Miguel de Sá Lima Magalhães Alves (A89468)
Alexandra de Barros Reigada (A84584)

ÍNDICE

| | | |
|------------|---|-----------------|
| 1. | <u>INTRODUÇÃO</u> | <u>3</u> |
| 2. | <u>DESCRIÇÃO DA ESTRUTURA DO PROJETO</u> | <u>3</u> |
| 2.1 | CATÁLOGO DE CLIENTES..... | 4 |
| 2.2 | CATÁLOGO DE PRODUTOS | 4 |
| 2.3 | GESTÃO DE FILIAIS..... | 4 |
| 2.4 | FATURAÇÃO GLOBAL | 4 |
| 2.5 | AUXILIAR | 5 |
| 3. | <u>DECISÃO</u> | <u>5</u> |
| 4. | <u>TESTES DE PERFORMANCE</u> | <u>5</u> |
| 5. | <u>CONCLUSÃO</u> | <u>7</u> |
| 6. | <u>GRAFO DE DEPENDÊNCIAS.....</u> | <u>7</u> |

1. Introdução

Foi-nos solicitado a realização deste projeto no âmbito da unidade curricular “Laboratórios de Informática III” que consiste na realização de uma aplicação que faz o processamento e gestão de vendas de 3 filiais.

Com o largo volume de dados com que fomos confrontados, tivemos que optar por técnicas de programação e algoritmos mais complexos para o tratamento destes.

Logo, neste projeto foram implementadas novas técnicas de programação, e visto que este era um projeto de maior escala, tivemos de recorrer a princípios como a modularidade e o encapsulamento.

Além disso, foram postos em prática conhecimentos previamente adquiridos em unidades curriculares anteriores, tais como Algoritmos e Complexidade, uma vez que para uma maior eficiência do programa necessitamos de analisar a complexidade das funcionalidades que o programa pedia.

2. Descrição da Estrutura do Projeto

Este projeto foi realizado com base em 3 ficheiros (mais 2 adicionais para testes de performance) sendo estes “Clientes.txt”, “Produtos.txt” e “Vendas_1M.txt” e tendo em conta o sistema MVC (Model-View-Controller).

O “Clientes.txt” é o ficheiro que fornece a informação dos clientes, “Produtos.txt” informação dos produtos e “Vendas_1M.txt” os registos de 1 milhão de vendas.

O **modelo** é constituído por 4 módulos e respetivos submódulos:

- **CatalogoCliente**
 - ↳ Cliente
- **CatalogoProduto**
 - ↳ Produto
- **Filiais**
 - ↳ FiliaisMeses
 - ↳ FiliaisClientes
 - ↳ FiliaisProdutos
- **Faturacao**
 - ↳ FaturacaoFiscal
 - ↳ FaturacaoMes
 - ↳ FaturacaoDados

A **vista** é constituída por um único módulo, UI.

O **controlador** é constituído pela main e o módulo SGV que comunicam com o modelo e com a vista.

2.1 Catálogo de Clientes

A estrutura de dados que constitui o catálogo de clientes é composta por um array de 26 posições que representam cada letra do alfabeto, por exemplo, a posição 0 corresponde à letra 'A' e por aí em diante.

Cada posição do array redireciona para uma árvore, em que cada nodo guarda uma string com a chave do cliente, balanceada por ordem alfanumérica.

2.2 Catálogo de Produtos

A estrutura de dados do catálogo de produtos é semelhante à dos clientes, apenas diferindo na string guardada, uma vez que o código dos produtos é diferente do código dos clientes.

2.3 Gestão de Filiais

A estrutura de dados que compõe as filiais já é um pouco mais complexa, uma vez que armazena a informação acerca das vendas o que nos obriga a utilizar uma estrutura diferente.

Esta é composta por um array de 3 posições, que correspondem às filiais em causa, em que cada posição nos indica outro array de 12 posições, correspondentes aos meses do ano. Em cada posição estará um apontador para uma árvore balanceada de clientes, que contém o código de cliente e um apontador para uma árvore balanceada de produtos.

Esta última árvore de produtos contém o código de produto e os restantes campos da venda (quantidade, tipo de compra e preço unitário) organizados num array dinâmico de estruturas.

2.4 Faturação Global

A estrutura de dados que compõe a faturação também terá a sua complexidade.

Esta é composta por uma árvore balanceada, contendo os códigos dos produtos. Cada nodo referencia um array de 3 posições cada uma associada a uma filial que por sua vez indica um array de 12 posições representando cada mês do ano, que contém um array de 2 posições (Primeira Posição – Compra normal, Segunda Posição – Compra em Promoção) que guarda uma estrutura de dados com as unidades e faturação associadas ao produto.

2.5 Auxiliar

Para poder responder eficientemente a certas queries tivemos a necessidade de criar 4 estruturas auxiliares: ProdSales, CodigosQuant, ArvoreQuant, ArvoreKey.

A estrutura ProdSales corresponde a um array dinâmico com dois parâmetros, a chave do produto e o respetivo lucro.

A estrutura CodigosQuant corresponde a um array dinâmico com dois parâmetros, a chave do produto e a respetiva quantidade.

A estrutura ArvoreQuant corresponde a uma árvore balanceada por quantidade contendo também informações adicionais como o array de códigos de produtos que foram comprados na mesma quantidade e o array de quantidades de clientes que compraram o respetivo produto.

A estrutura ArvoreKey corresponde a uma árvore balanceada alfanumericamente pelo código de produto, dando informação também da quantidade que foi comprada, número de clientes que compraram este produto e os códigos dos clientes que o compraram.

3. Decisão

Depois de uma análise minuciosa das queries que nos foram apresentadas, apercebemo-nos que grande parte destas exigia procura nas estruturas de dados, então pensámos em duas alternativas: tabelas de hash ou árvores balanceadas ($O(1)$ e $O(\log(n))$, respetivamente).

Por fim, optámos pelas árvores balanceadas uma vez que o varrimento destas é mais eficiente ($O(n)$) do que tabelas de hash ($O(2n)$).

Como certas queries implicam o acesso a informação por filiais, por meses e/ou por letra decidimos organizar estes parâmetros em arrays de 3,12 e 26 posições, respetivamente, para facilitar a obtenção destes dados.

Para aumentar ainda mais a eficiência recorreremos a estruturas auxiliares supracitadas dependendo das queries atendendo aos requerimentos de cada uma.

4. Testes de Performance

| Query | Valor | Valor global | Por filial | Jan a Dez | Jan | Limites | | | | | |
|-------|----------|--------------|------------|-----------|----------|---------|----------|---------|----------|----------|----------|
| | | | | | | 1 | 10 | 100 | 1000 | 10000 | 200000 |
| 1 | 4,053209 | - | - | - | - | - | - | - | - | - | - |
| 2 | 0,02047 | - | - | - | - | - | - | - | - | - | - |
| 3 | - | 0,007844 | 0,007653 | - | - | - | - | - | - | - | - |
| 4 | - | 0,059399 | 0,189173 | - | - | - | - | - | - | - | - |
| 5 | 0,028049 | - | - | - | - | - | - | - | - | - | - |
| 6 | 0,06918 | - | - | - | - | - | - | - | - | - | - |
| 7 | 0,000119 | - | - | - | - | - | - | - | - | - | - |
| 8 | - | - | - | 0,03086 | 0,026135 | - | - | - | - | - | - |
| 9 | 0,039194 | - | - | - | - | - | - | - | - | - | - |
| 10 | 0,000029 | - | - | - | - | - | - | - | - | - | - |
| 11 | - | - | - | - | - | 0,37413 | 0,349766 | 0,38619 | 0,669053 | 0,726056 | 1,567738 |
| 12 | - | - | - | - | - | 0,00012 | 0,00337 | 0,00396 | 0,00327 | 0,00402 | 0,00056 |
| 13 | 0,000035 | - | - | - | - | - | - | - | - | - | - |

FIG.1 Tabela de tempos de execução em segundos para o ficheiro “Vendas_1M”

| Query | Valor | Valor global | Por filial | Jan a Dez | Jan | Limites | | | | | |
|-------|-----------|--------------|------------|-----------|---------|----------|----------|----------|----------|----------|----------|
| | | | | | | 1 | 10 | 100 | 1000 | 10000 | 200000 |
| 1 | 13,394485 | - | - | - | - | - | - | - | - | - | - |
| 2 | 0,002379 | - | - | - | - | - | - | - | - | - | - |
| 3 | - | 0,0203332 | 0,016917 | - | - | - | - | - | - | - | - |
| 4 | - | 0,046655 | 0,141677 | - | - | - | - | - | - | - | - |
| 5 | 0,023152 | - | - | - | - | - | - | - | - | - | - |
| 6 | 0,055768 | - | - | - | - | - | - | - | - | - | - |
| 7 | 0,000152 | - | - | - | - | - | - | - | - | - | - |
| 8 | - | - | - | 0,036839 | 0,03348 | - | - | - | - | - | - |
| 9 | 0,063458 | - | - | - | - | - | - | - | - | - | - |
| 10 | 0,000062 | - | - | - | - | - | - | - | - | - | - |
| 11 | - | - | - | - | - | 0,781143 | 0,723894 | 0,965279 | 1,214713 | 1,870201 | 4,34271 |
| 12 | - | - | - | - | - | 0,000297 | 0,000286 | 0,000258 | 0,000870 | 0,000495 | 0,000988 |
| 13 | 0,000033 | - | - | - | - | - | - | - | - | - | - |

FIG 2. Tabela de tempos de execução em segundos para o ficheiro “Vendas_3M”

| Query | Valor | Valor global | Por filial | Jan a Dez | Jan | Limites | | | | | |
|-------|-----------|--------------|------------|-----------|----------|----------|----------|----------|----------|----------|----------|
| | | | | | | 1 | 10 | 100 | 1000 | 10000 | 200000 |
| 1 | 23,482487 | - | - | - | - | - | - | - | - | - | - |
| 2 | 0,002360 | - | - | - | - | - | - | - | - | - | - |
| 3 | - | 0,020071 | 0,020868 | - | - | - | - | - | - | - | - |
| 4 | - | 0,036199 | 0,094551 | - | - | - | - | - | - | - | - |
| 5 | 0,021467 | - | - | - | - | - | - | - | - | - | - |
| 6 | 0,041527 | - | - | - | - | - | - | - | - | - | - |
| 7 | 0,000161 | - | - | - | - | - | - | - | - | - | - |
| 8 | - | - | - | 0,039148 | 0,036239 | - | - | - | - | - | - |
| 9 | 0,059988 | - | - | - | - | - | - | - | - | - | - |
| 10 | 0,000056 | - | - | - | - | - | - | - | - | - | - |
| 11 | - | - | - | - | - | 0,731526 | 1,030985 | 1,407820 | 1,907704 | - | - |
| 12 | - | - | - | - | - | 0,00128 | 0,000316 | 0,001320 | 0,001348 | 0,001401 | 0,001603 |
| 13 | 0,000035 | - | - | - | - | - | - | - | - | - | - |

FIG 3. Tabela de tempos de execução em segundos para o ficheiro “Vendas_5M”

A partir destas tabelas podemos observar os tempos de execução, query a query, dos 3 ficheiros de vendas fornecidos.

Analizamos que a query 11 é, excetuando o carregamento dos ficheiros (query 1), a que demora mais tempo para executar, uma vez que esta query não é favorecida, de todo, pelo modo como está organizada a nossa estrutura de dados. Isto porque tem que varrer as estruturas de faturação e de filiais e, para se obter os dados do produto, é preciso percorrer cliente a cliente, não conseguindo acesso direto através da estrutura das filiais.

Como esperado, os tempos de execução aumentam com o aumento de volume de dados e com o aumento dos limites.

5. Conclusão

Este projeto excedeu as nossas expectativas, uma vez que conseguimos cumprir todos os objetivos propostos pela equipa docente e obter resultados bastante satisfatórios.

Acreditamos que fosse possível melhorar os tempos de execução, especialmente do carregamento de ficheiros, e a apresentação da interface do usuário.

Também detetamos uma falha no nosso código para a query 11 com limites muito grandes (aproximadamente a partir do 10.000), mas apenas com o ficheiro de “Vendas_5M”, que não conseguimos corrigir.

Dito isto, apesar desta falha e de possíveis melhorias que poderiam ser feitas, conseguimos ultrapassar os obstáculos impostos e concluir o projeto proposto.

6. Grafo de Dependências

