

Faculdade de Engenharia da Universidade do Porto

MIEIC - Algoritmos e Estruturas de Dados

Relatório trabalho prático - Parte 2

Cadeia de Farmácias

08/01/2018

Turma 5 – grupo 1

David Luís Dias da Silva – up201705373@fe.up.pt

Gaspar Santos Pinheiro – up201704700@fe.up.pt

Luís Pedro Pereira Lopes Mascarenhas Cunha – up201706736@fe.up.pt

Índice

| | |
|---------------------------------|----------|
| Descrição do tema | 3 |
| 1ª parte | 3 |
| 2ª parte | 3 |
| Solução implementada | 4 |
| Casos de utilização | 6 |
| Dificuldades encontradas | 8 |
| Esforço dedicado | 8 |

Descrição do tema

1ª parte

O tema escolhido de entre os propostos para o trabalho prático consiste na realização de um programa para gerir uma cadeia de farmácias, fornecendo ao utilizador funcionalidades básicas que lho permitam. Os principais elementos da cadeia de farmácias são as farmácias, os clientes, os empregados e as vendas (nas quais receitas dos pacientes podem ser aviadas).

As principais funcionalidades presentes no programa permitem lidar com estes elementos e são realizar vendas, alterar dados referentes aos empregados e clientes da cadeia, adicionar farmácias à cadeia e gerir o seu *stock*, bem como consultar dados relativos a todos os elementos da cadeia.

Outra funcionalidade é a de ser possível gerir diferentes cadeias (em utilizações diferentes do programa) e guardar os dados relativos à cadeia em ficheiros de texto para utilização futura.

2ª parte

A segunda parte tem como objetivo complementar o que foi desenvolvido na primeira usando estruturas não lineares que foram lecionadas nas aulas, em particular, as árvores binárias de pesquisa (BST), as tabelas de dispersão (Hash Tables) e as filas de prioridade (Heap).

A integração destas no trabalho, implicaria o uso de operações de criação, leitura, atualização e eliminação.

Solução implementada

A adição de uma BST ao projeto não exigiu muitas alterações no mesmo, dada a organização prévia do nosso trabalho. Foi necessária uma pequena pesquisa inicial para encontrar uma forma de ter uma estrutura de dados set, da STL, constituída por apontadores para objetos da classe Cliente.

Um problema encontrado foi como fazer listagens várias com critérios a definir pelo utilizador, já que não é possível alterar a ordem de um set. A solução encontrada foi manter um vetor de apontadores para objetos da classe Cliente a par com o set, isto é, sempre que é adicionado ou removido um cliente ao set, a operação também é realizada no vetor. Quando o utilizador desejar consultar o resumo da lista de clientes, o vetor é ordenado conforme o seu pedido e, de seguida, é mostrado no ecrã. Desta forma a ordem original do set é sempre mantida.

Foi, ainda, necessário um foco em como tirar partido da pesquisa logarítmica das BSTs. Esta vantagem é utilizada sempre que o utilizador desejar consultar um cliente, devendo intrudizir o nome do cliente, distrito e NIF.

A adição de filas de prioridade ao projeto foi efetuada acrescentando duas novas classes ao projeto e integrando-as nas cadeias e farmácias existentes. As novas classes adicionadas foram a classe “Fornecedor” e a classe “Encomenda”.

Uma farmácia possui agora uma fila de prioridade com os seus produtos, prioridade essa que é definida pela quantidade existente em *stock* desses mesmos produtos. Isto implica que para uma farmácia adicionar produtos ao seu stock deve começar por adicionar os produtos que existem em menor quantidade, informação que obtém através de uma fila de prioridade.

Assim, é agora dada a opção ao utilizador de “repor o *stock*” de uma farmácia, sendo-lhe mostrado os produtos que possuem uma quantidade inferior a N (informação que é obtida, mais uma vez, através de uma fila de prioridade) e podendo decidir qual será a nova quantidade pretendida desses produtos.

A adição de produtos a farmácias é agora feita através de encomendas a um fornecedor ou mais fornecedores.

A classe “Fornecedor” suporta dois tipos, fornecedor de medicamentos ou de produtos genéricos. Para que seja possível adicionar novos produtos a uma farmácia esta tem primeiro de adicionar um fornecedor de cada tipo, no entanto, não existe limite para a quantidade de fornecedores diferente que uma farmácia pode ter. Os fornecedores são guardados na cadeia através de vetores e através de filas de prioridade nas farmácias.

A adição de novos produtos leva à criação de encomendas entre a farmácia e os fornecedores, sendo que a adição de medicamentos gera encomendas entre a farmácia e um fornecedor de medicamentos e a adição de produtos gera encomendas entre a farmácia e um fornecedor de produtos.

As encomendas são representadas através da classe “Encomenda” que guarda informações acerca dos produtos que contém a encomenda, a data e hora da sua realização, dos intervenientes (uma farmácia e um fornecedor) e sobre o seu valor.

De modo a dar mais ênfase à estrutura fila de prioridade, numa farmácia, os fornecedores são guardados em duas filas de prioridade (uma para fornecedores de medicamentos e outra para fornecedores de produtos genéricos). Estas filas são usadas para escolher que fornecedores executam as encomendas que vão sendo feitas pela farmácia. A prioridade é definida pela quantidade total de encomendas que foram satisfeitas pelo fornecedor (menos encomendas, maior prioridade), medida que visa a simular um comércio “justo” entre as farmácias e os fornecedores.

A adição de tabelas de dispersão para manter registo dos empregados, atuais e antigos, da farmácia, levou a que a estrutura utilizada para guardar os empregados fosse mudada para um unordered_set da STL, quer na classe cadeia, onde é mantida uma tabela de dispersão com apontadores para todos os empregados da cadeia, incluindo os que não estão atualmente ligados à cadeia, quer na classe farmácia, onde cada farmácia tem uma tabela de dispersão com apontadores para os empregados que lá trabalham.

De forma a manter registo da ligação anterior dos empregados à cadeia, foram adicionados atributos à classe Empregado de forma a ser possível guardar o número de meses em que o empregado esteve ligado à farmácia, bem como a saber se ele está atualmente ligado à farmácia. Foram ainda adicionados métodos que permitem despedir e recontratar empregados que já possuem ligação à cadeia.

Foram acrescentados os atributos *ultimaDataContratacao* (representa a última data em que o empregado efetuou uma nova ligação à cadeia), *ultimaDataDespedimento* (representa a data em que o empregado terminou a última ligação à cadeia, caso o empregado esteja atualmente ligado à cadeia, este atributo tem o valor 00-00-0000, o que permite saber se o empregado está ou não atualmente ligado) e *mesesLigacaoAnterior* (representa o número de meses total das ligações anteriores do empregado à cadeia de farmácias, excluindo a atual). Sempre que um empregado é despedido, o nome da farmácia à qual ele está ligado é substituído por uma *string* vazia, o atributo *ultimaDataDespedimento* é atualizado para a data atual e o atributo *mesesLigacaoAnterior* é atualizado com o número de meses que a ligação atual durou.

Casos de utilização

O programa inicia dando duas hipóteses ao utilizador, criar uma nova cadeia de farmácias ou carregar os dados de uma cadeia de farmácias. Para a segunda opção, o utilizador deve indicar o nome da cadeia que pretende continuar a gerir.

De seguida, no menu principal, o utilizador tem cinco opções, que consistem em gerir clientes, empregados, farmácias e fornecedores e realizar vendas.

Dentro da funcionalidade realizar venda, o utilizador deve especificar a farmácia na qual a venda está a ser realizada, tal como o empregado responsável. O utilizador deve ainda indicar o cliente, devendo criar uma ficha de cliente para o caso deste ainda não existir. Depois de especificados estes dados, o utilizador pode adicionar produtos e receitas à venda, bem como consultar o “saco de compras” até ao momento. À medida que produtos e receitas são adicionados à venda, a possibilidade do *stock* da farmácia satisfazer a procura é verificada. Para terminar, é apresentado ao utilizador o custo da venda, de modo a este proceder à confirmação.

A funcionalidade gerir farmácias permite:

- Ver um resumo geral das farmácias existentes na cadeia.
- Consultar informações relativas a cada farmácia, como os empregados, vendas, produtos e fornecedores. Nesta opção, também é possível consultar a prioridade de encomendas que devem ser efetuadas (informação obtida através da fila de prioridade).
- Adicionar farmácias novas à cadeia, sendo para isso necessário criar um gerente para a nova farmácia.
- Gerir o *stock* de cada farmácia, permitindo adicionar produtos e remover produtos.
- Agora, adicionar produtos a individualmente a uma farmácia apenas é possível se não existir nenhum produto com maior prioridade, isto é, menor quantidade em stock. Adicionar produtos gera encomendas ao fornecedor prioritário (com menor número de encomendas satisfeitas) da farmácia que fornece o tipo de produtos adicionados (genéricos ou medicamentos).
- Existe agora também a opção de “Repor stock” que permite ao utilizador reabastecer todos os produtos com quantidade inferior a N, definindo também a nova quantidade. Esta ação gera uma ou duas encomendas, uma para cada tipo de produto que contem a reposição (produtos ou medicamentos). As encomendas são satisfeitas pelo fornecedor prioritário de cada tipo.
- Alterar o gerente de uma farmácia.
- Adicionar e remover fornecedores das farmácias. É de notar que agora, para poder adicionar produtos uma farmácia necessita de ter um fornecedor de cada tipo (medicamentos e produtos).

A funcionalidade gerir clientes permite:

- Ver uma listagem com informações dos clientes da cadeia, ordenados por um critério à escolha pelo utilizador.
- Gerir um cliente em específico, onde, após indicar o cliente que pretende gerir, pode alterar o seu distrito de residência, alterar a sua morada, consultar as suas compras, bem como removê-lo da lista de clientes.

A funcionalidade gerir empregados permite:

- Ver uma listagem com informações dos empregados da cadeia, ordenados por um critério à escolha pelo utilizador.
- Consultar empregados que não têm uma ligação atual à cadeia, ordenados por um critério à escolha do utilizador, incluindo o número de meses em que este esteve ligado à cadeia.
- Gerir um empregado, podendo alterar informações relativamente a este, tal como o cargo, salário, morada e farmácia onde trabalha.
- Adicionar novos empregados à cadeia.
- Recontratar empregados que têm ligações antigas com a cadeia.
- Despedir um empregado atualmente ligado à cadeia.

A funcionalidade de gerir fornecedores permite:

- Ver uma listagem do resumo dos fornecedores ordenada pela forma escolhida pelo utilizador (nome, número de encomendas, tipo de fornecedor).
- Adicionar um novo fornecedor à cadeia. (a inexistência da opção de remover fornecedores é deliberada, pois serve como modo de guardar a informação permanentemente acerca de todas as encomendas efetuadas).
- Consultar fornecedores da cadeia. Nesta opção o utilizador escolhe um fornecedor (fornecedores são identificados pelo seu nome, sendo este único), e poderá ver o resumo das suas encomendas, ou consultar encomendas especificamente por data de realização da encomenda ou pela farmácia que a originou.
- As encomendas contêm informação acerca do valor, lista de produtos, data e hora de realização e sobre os seus intervenientes.

Dificuldades encontradas

As principais dificuldades sentidas no trabalho foram o facto de este ser construído sobre o que foi feito anteriormente. Isto leva a que seja preciso ter consciência do que as novas adições implicam para o que foi desenvolvido anteriormente e sobre a melhor forma para abordar o problema de modo a evitar redundâncias e a impedir que as novas adições criem erro no código desenvolvido anteriormente. Este facto também veio por à prova a documentação das funções e estruturas que foi feita para a primeira parte, na medida em que afetava diretamente a velocidade de desenvolvimento e da integração das novas funcionalidades no projeto.

Esforço dedicado

O esforço dedicado pelos intervenientes no trabalho foi equiparável, pois inicialmente procedemos à divisão de tarefas e todos os membros cumpriram os seus deveres. Não obstante desta divisão, a construção do projeto não foi efetuada de forma estanque, tendo os membros do grupo contribuíram de uma forma proativa para a resolução do problema, cooperando e discutindo abordagens, soluções e melhorias. No entanto, consideramos que o trabalho não está perfeito, e que há espaço para uma otimização e correção de eventuais problemas.