

[AEDA]-Algoritmos e Estruturas de Dados

Empresa de Táxis



Turma 4 – Grupo I

200800621 – Luís Pedro Moras
201200764 – Pedro Franca Santos

Descrição do Tema

O nosso trabalho incide na criação de um sistema de gestão dos serviços de uma empresa de táxis.

A empresa possui uma carteira de clientes que podem particulares ou empresariais que precisam de ser registrados no sistema e mantem um histórico de serviços requisitados, no entanto também presta serviços a clientes ocasionais.

Os serviços prestados (viagens) tem uma quantidade de tem uma quantidade de atributos (Origem, Destino, Valor, Data, Duração) que também devem ser arquivados em associação com o cliente que os requisitou.

Todos os serviços dispõem de varias formas de pagamento. As formas de pagamento disponíveis diferem com o tipo de cliente que requer o serviço. Algumas delas tem taxas agravadas.

A empresa pode também optar por realizar promoções aos seus clientes registrados por meio de um cartão de pontos.

Por fim deve ser possível gravar o estado atual da empresa para um ficheiro como também utilizar um ficheiro previamente gravado para inicializar uma empresa com um estado anterior.

////////////////////////////////////

A empresa cria agora faturas(nova classe) de cada vez que um serviço for registado. As faturas são guardadas numa **arvore de pesquisa binaria**(BST.h).

Adicionamos também a funcionalidade de reservar serviços futuros(em relação a data e hora atual do computador). Essas reservas são guardadas numa **fila de prioridade** em que no topo da fila encontra-se a viagem que vai ser realizada mais cedo.

Neste momento a empresa também classifica como inativos clientes que não usem o serviço há trinta ou mais dias. Estes clientes passam então a ser guardados numa **tabela de dispersão voltando** a ser classificados como clientes ativos e retirados da mesma tabela quando realizarem um novo serviço.

Solução Implementada

No implementação do nosso trabalho sobre o tema cobrido anteriormente o nosso grupo Implementou 3 Classes principais “Firma”, “Cliente” e “Servico” que seguem uma estrutura do tipo FIRMA<-CLIENTES-<SERVICO pois a Classe Firma possui um vector de apontadores para Clientes e a Classe Cliente possui um vector de Servicos.

A classe base Cliente possui também como classes derivadas ClienteParticular e ClienteEmpresarial.

Todas estas classes possuem funções básicas como set’s, get’s , sort’s e outras que achamos necessárias para a resolução do problema.

Todas estas funções são chamadas num conjunto de menus que permitem ao utilizador tomar decisões e introduzir informação.

////////////////////////////////////

Algumas das soluções implementadas diferem ligeiramente nas sugeridas no guião da segunda parte do trabalho devido a estrutura da primeira parte do nosso trabalho não ser completamente compatível com as implementações pedidas, no entanto consideramos que os ligeiros desvios as implementações pedidas que implementamos permitem avaliar os mesmos *know-hows*.

No caso de implementação da BST criamos uma nova classe **Fatura** em que cada vez que um serviço for criado também se emite uma nova fatura, todos os elementos da fatura (nome do cliente, nif, valor da fatura, etc) são usados os mesmo que na criação do serviço, a única diferença sendo que a fatura possui um numero de fatura (static int) que vai sendo incrementado a cada nova fatura emitida.

A nossa implementação da BST difere da pedida pois as faturas não estão ordenadas por ordem alfabética mas sim por ordem crescente de nif (pois no

nosso trabalho o nif funciona como o id do cliente ao invés do nome, por ser possível na nossa aplicação um cliente registado alterar o seu nome e morada mas nunca sendo possível alterar o nif, como na realidade) no entanto foi igualmente preciso fazer overload do operador < para a classe Cliente e a BST ficou então ordenada por ordem crescente de nif mantendo as faturas do mesmo cliente(mesmo nif) juntas e ordenadas por ordem de data como era pedido.

Também para a implementação de **filas de prioridade** visto não termos viaturas no nosso projecto decidimos então implementar uma nova funcionalidade de reservar viagens para o futuro (o programa usa a data real do computador) e permite registo de viagens que já ocorreram e reservar viagens para o futuro guardando-as numa **fila de prioridade** em que no topo da pilha encontra-se a viagem a ser realizada mais proximamente (usando uma struct para comparação e overload de operadores para data e hora da reserva com tempo real). Quando chega a hora da viagem que esta no topo da esta automaticamente sai da fila e entra no vector de Servicos do Cliente.

Na implementação da **tabela de dispersão** tivemos a abordagem de criar uma **tabela de dispersão** para guardar clientes inativos (são considerados clientes inativos clientes que não usem o serviço por um tempo igual ou superior a 30 dias) quando um cliente inativo volta a utilizar um serviço este vai então ser retirado da tabela de dispersão e colocado de volta no vector de Clientes da Firma.

Casos de Utilização

Esta aplicação foi desenvolvida com o intuito de ser uma plataforma digital utilizada para gestão de uma empresas de táxis.

Dificuldade Encontradas

Por vezes vimo-nos deparados com algumas situações relacionadas com vectores de apontadores e também com a leitura dos ficheiros que nos consumiram algum tempo a contornar e a arranjar uma solução. No entanto este tipo de situações são naturais e expectáveis num trabalho de programação deste género e no nosso curso em geral. Sentimos também alguma dificuldade na criação do UML.

Aparte disso a única dificuldade (que não será bem classificada como dificuldade) foi que apenas em cima do prazo de entrega nos apercebemos que por lapso não havíamos usado exceções tendo no entanto todo o programa com mecanismos de controlo a base de if's, substituímos algumas destas estruturas de controlo por exceções por questões de avaliação, no entanto, infelizmente, o tempo não nos permitiu alterar todas.

////////////////////////////////////

Tal como referido para a primeira parte do trabalho a medida que íamos avançando no trabalho fomos nos deparando com algumas dificuldades intrínsecas a toda a aprendizagem, quer por vezes na lógica (estruturamento do raciocínio) e por outras vezes na própria parte de sintaxe da linguagem.

No entanto podemos afirmar que a maior dificuldade terá sido mesmo a falta de tempo (por ventura por má gestão da nossa parte) em conciliação com a restante carga de trabalhos e avaliações das restantes unidades curriculares.

Participação

Apesar de logo desde a primeira parte do trabalho termos perdido um elemento do grupo (que desistiu da cadeira) tendo isto refletindo-se numa maior carga de trabalho para os elementos restantes, todos os membros do grupo trabalharam em igual percentagem e conseguiram com mais ou menos dificuldade concluir o mesmo.