

Trabalho 1 - Gestão de um Hipermercado (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- O bom funcionamento do hipermercado requer a existência programada de serviços de manutenção e limpeza. Por vezes, surge a necessidade imediata de efetuar estes serviços, sem escalonamento prévio. Todos os serviços programados (já realizados ou não) e serviços imediatos de manutenção e limpeza são armazenados numa **árvore binária de pesquisa**, sendo a ordenação efetuada por data/hora do serviço. Um serviço é descrito por estado (já realizado ou não). Deve ser permitido: adicionar serviços programados ou imediatos, remover serviços (porque foram realizados há mais de uma semana, por exemplo), alterar o estado de um serviço (de não realizado para realizado). Devem ainda ser permitidas listagens várias.
- Os clientes inativos (em que o último movimento foi efetuado há mais de 1 ano) devem ser guardados numa **tabela de dispersão**. Deve ser permitido a inserção de clientes na tabela de dispersão, remoção de clientes da tabela de dispersão (porque se pretende eliminar qualquer referência ao cliente) ou alteração de dados desses clientes (contacto, por exemplo).
- Quando o stock de um produto atinge um nível mínimo especificado, este deve ser reposto e são efetuadas compras desse produto ao fornecedor respetivo. Considere uma **fila de prioridade** que guarda informação sobre fornecedores. Um fornecedor é caracterizado pela sua denominação, localização e produtos que vende. O sistema deve encontrar o fornecedor mais próximo que possui o produto que o hipermercado pretende comprar. Deve ser permitido: comprar produtos a um fornecedor (alterando o stock quer do hipermercado, quer do fornecedor); adicionar novos fornecedores com produtos associados; remover fornecedores (porque este deixou de existir).

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 2 - Gestão de uma Escola de Condução (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere que uma escola de condução é caracterizada, para além da informação já considerada na primeira parte do trabalho, por: designação, localização e número máximo de alunos. Se um aluno se pretende inscrever numa escola que está completa (não possui vagas), deve ser possível visualizar quais as outras escolas existentes. Considere que todas as escolas de condução são armazenadas numa **árvore binária de pesquisa**, sendo a ordenação efetuada por localização ou capacidade de alunos. Deve ser permitido: adicionar escolas de condução, remover escolas de condução (porque cessaram a atividade), alterar a informação relativa a uma escola de condução (aumentou/diminuiu o nº máximo de alunos que aceita, por exemplo). Devem ainda ser permitidas listagens várias.
- Os alunos inativos (que deixaram de ter aulas há mais de 1 ano) devem ser guardados numa **tabela de dispersão**. Para efeitos de publicidade, interessa manter a informação atualizada de antigos alunos da escola. Deve ser permitido a inserção de alunos na tabela de dispersão, remoção de alunos da tabela de dispersão (porque não interessa manter a referência ao aluno) ou alteração de dados desses alunos (contacto, por exemplo).
- Considere uma **fila de prioridade** que guarda informação sobre oficinas automóveis. Uma oficina é caracterizada pela sua denominação, localização, marcas de automóveis em que é especializada e disponibilidade (nº dias até que possa efetuar o serviço). Um automóvel da escola de condução pode necessitar de a assistência de oficina para serviços específicos (substituição de uma peça,...) ou serviços usuais (mudança de óleo,...). O sistema deve indicar a oficina disponível mais cedo: i) que é especializada na marca do veículo em causa, se se tratar de um serviço específico, ou ii) qualquer oficina, no caso de se tratar de um serviço normal. Sempre que uma oficina aceita um serviço, a sua disponibilidade diminui. Considere também a ocorrência de término de um serviço, caso em que a disponibilidade da oficina aumenta. Deve ser permitido: adicionar novas oficinas, alterar as marcas em que uma oficina está especializada; remover oficina (porque a oficina cessou a atividade).

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 3 - Gestão de turmas numa Escola (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Os funcionários da escola estão alocados a espaços específicos, sendo também responsáveis pela supervisão das turmas que habitualmente ocupam esses espaços. Um funcionário é caracterizado pelo nome, contacto e turmas que supervisiona. Os funcionários são armazenados numa **árvore binária de pesquisa**, sendo a ordenação efetuada por nome. Deve ser permitido: adicionar funcionários, remover funcionários (ficando as suas turmas sem supervisão), alterar a informação de um funcionário (alterar as turmas que supervisiona, por exemplo). Devem ainda ser permitidas listagens várias.
- Os professores que deixaram de lecionar na escola, devem ser guardados numa **tabela de dispersão** para futuro contacto, no caso de ser necessário contratar novos docentes. A cada professor deve estar associada informação sobre nome, contacto telefónico e disciplinas da sua área. Deve ser permitida a inserção de professores na tabela de dispersão, remoção de professores da tabela de dispersão (porque este professor passa à reforma, por exemplo) ou alteração de dados desses professores (contacto, por exemplo).
- Para ajudar os alunos na compra de livros escolares, a escola dispõe de informação sobre livrarias que vendem livros de determinadas áreas. Considere uma **fila de prioridade** que guarda informação sobre as livrarias. Uma livraria é caracterizada pela sua denominação, localização, área de especialidade (conjunto de disciplinas a que pertencem os livros que vende) e anos de escolaridade (anos de escolaridade a que pertencem os livros que vende). O sistema deve encontrar a livraria mais próxima que vende o(s) livro(s) que o aluno pretende adquirir. O aluno pode especificar ou não quais as disciplinas dos livros que pretende, e o sistema procura a livraria mais próxima dessas áreas de especialidades (no primeiro caso) ou simplesmente a livraria mais próxima (no segundo caso). Deve ser permitido: adicionar livrarias; alterar informação sobre livrarias (alterar áreas de especialidade ou anos de escolaridade); remover livrarias (porque a livraria cessou a atividade).

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 4 – Gestão de uma Galeria d’Arte (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Para melhorar a gestão do acervo da galeria, resolveu-se guardar as informações relativas às peças d’arte numa **árvore binária de pesquisa**, sendo a ordenação realizada pelo ano da obra (as mais antigas em primeiro lugar), pelo tipo, pelo autor e pelo proprietário. Deve ser permitido: adicionar novas obras ao acervo, remover obras do acervo (porque deixaram de pertencer à galeria, estarem expostas ou guardadas no seu depósito, por exemplo), alterar informações da obra (como o seu proprietário atual, entre outras informações). Devem ainda ser permitidas listagens várias.
- Para efeitos de publicidade e promoção de leilões, a galeria passou a guardar numa **tabela de dispersão** todos os seus clientes inativos, que já não realizam quaisquer tipos de transações com a galeria (compra/venda de peças de arte, exposições, etc.) a um determinado período (por exemplo 12 meses). Interessa portanto guardar a morada e contactos atualizados dos clientes inativos. Deve ser permitido a adição de antigos clientes, remover registos dos clientes (tão logo passem a realizar uma próxima transação de compra e venda ou exposição), ou alterar os seus dados, como morada, a fim de se poder realizar publicidade endereçada, etc.
- Como o número de clientes da galeria interessados em utilizar o espaço de exposições aumentou, a galeria resolveu realizar marcações seguindo alguns critérios de prioridade. Assim, as marcações do espaço de exposições são guardadas numa **fila de prioridade**, onde clientes frequentes (com maior número de transações realizadas nos últimos 12 meses), assim como o valor total acumulado das transações são os principais critérios de prioridade. Deve ser permitido: adicionar novos pedidos de reserva do espaço de exposições, cancelar reservas já realizados, ou alterar informações da reserva (como tipo de exposição, espaço necessário, lista de peças a serem expostas, e período da exposição).

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 5 – Empresa de Entrega ao Domicílio de Restaurantes (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- No sentido de organizar melhor as informações relativas aos restaurantes para os quais a empresa oferece o serviço de entrega ao domicílio, passou a guardar numa **árvore binária de pesquisa** os menus do dia, de cada restaurante, organizados pelo preço médio do menu do dia e nome do restaurante que o oferece (o menu do dia típico inclui bebida, prato principal e sobremesa, podendo ser fixo durante a semana ou alterado diariamente). É importante notar que nem todos os restaurantes que trabalham com a empresa oferecem este item. Deve ser permitido: adicionar novos menus, remover menus, alterar a informação relativa ao preço do menu, assim como a sua descrição. Devem ainda ser permitidas listagens várias.
- Para efeitos de publicidade, a empresa passou a guardar numa **tabela de dispersão** os clientes inativos (que já não realizam encomendas há mais de um ano, por exemplo). Interessa portanto guardar a morada atualizada dos clientes, com o seu NIF para efeitos de emissão de fatura, assim como a lista das suas preferências (últimos pedidos ou pedidos mais frequentes). Deve ser permitido a adição de antigos clientes, remover registos dos clientes (tão logo passem a realizar um próximo pedido), ou alterar os seus dados, como morada, a fim de se poder realizar publicidade endereçada, etc.
- À medida que novos pedidos são realizados, estes são colocados numa **fila de prioridade** a fim de serem processados. A ordenação dos pedidos na fila de prioridade deve obedecer aos seguintes critérios, nesta ordem: clientes frequentes (a partir do número de pedidos realizados nos últimos 12 meses); valor do pedido (valores mais elevados são prioritários); e menor distância do ponto de coleta do pedido ao ponto de entrega. Deve ser permitido: adicionar novos pedidos, cancelar pedidos realizados e ainda não processados, ou alterar pedidos não processados (retirando ou adicionando itens, com a respetiva atualização do seu valor).

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 6 - Gestão de Condomínios (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere agora que a empresa que gere os condomínios considerou mais lucrativo adjudicar os serviços de manutenção a empresas especializadas, para serviços que possam custar acima de um valor previamente previsto no orçamento do condomínio para essa finalidade. Caso o orçamento do serviço de manutenção ultrapasse o valor estipulado para este tipo de serviço, o condomínio então contratará o serviço a empresas especializadas, cujos contactos e informações guarda numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada por localização da empresa relativamente ao condomínio que solicitou o serviço e o último preço pago para aquele tipo de serviço. Deve ser permitido: adicionar empresas prestadoras de serviços especializados, remover empresas (porque mudaram a sua especialidade ou por se tornarem muito caras, por exemplo), alterar a informação relativa ao último serviço prestado (valor pago, tipo de serviço, condomínio onde o serviço foi realizado, etc.). Devem ainda ser permitidas listagens várias.
- Os antigos condóminos (que deixaram de pertencer a um dos condomínios geridos pela empresa) devem ser guardados numa **tabela de dispersão**. Para efeitos de auditoria, interessa manter a informação atualizada dos antigos condóminos; um antigo condómino poderá também voltar a ser condómino ativo, caso volte a adquirir outro imóvel de um dos condomínios da empresa. Deve ser permitido a inserção de antigos condóminos na tabela de dispersão, remover registos da tabela (por ter ultrapassado, para efeitos legais, o período de tempo que a informação necessita ser guardada pela empresa gestora) ou alterar os dados do antigo condómino, para efeitos de contacto, por exemplo (como morada atualizada).
- Considere agora que os condomínios passaram a ter um salão de eventos (para realização de festas, reuniões, ou outras atividades que queiram realizar os condóminos). A utilização do salão de eventos requer marcação prévia, sendo que a sua reserva para um determinado dia é realizada respeitando a prioridade dos condóminos que pagam os maiores valores de condomínio. Os agendamentos, por dia, devem ser armazenados numa **fila de prioridade**, que requer a confirmação um determinado prazo, o mais tardar. Caso o condómino com maior prioridade não confirme no prazo estipulado, a sua marcação caduca, passando o seguinte a ter que confirmar o seu interesse. O prazo deve diminuir com a proximidade da data do evento. Deve ser permitido: adicionar novas reservas, alterar uma reserva, caso o condómino possua imóvel noutra condomínio e deseje modificar o local do evento (entre salões dos condomínios onde possui fração), ou mesmo remover marcações.

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 7 – Empresa de Traduções (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- A empresa decide, de tempos em tempos, contratar novos tradutores. De modo a acelerar o processo de contratação a empresa mantém uma lista de tradutores interessados em trabalhar para a empresa. A informação sobre os candidatos é mantida numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pelo nome do tradutor. A informação guardada para os candidatos a tradutores é a mesma dos tradutores da empresa. Deve ser permitido adicionar, remover, e alterar os candidatos a tradutores. Devem ainda ser permitidas listagens várias.
- A empresa decidiu guardar a informação sobre as encomendas já terminadas. Este registo histórico dever ser guardado numa **tabela de dispersão**. Deve ser permitido, na tabela de dispersão: a inserção de encomendas já terminadas; remoção de encomendas (que são anteriores a determinada data limite); ou alteração de encomendas na tabela.
- Considere que os textos a serem traduzidos devem agora ser atendidos, não pela ordem de chegada, mas dando mais prioridade aos textos que se traduzem mais rapidamente. Para isso, mantenha a carteira de encomendas numa **fila de prioridade**. Deve ser possível inserir, remover e alterar as encomendas guardadas na fila de prioridade.

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 8 – Gestão de um Bloco Operatório (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- As existências de material utilizado nas cirurgias são guardadas numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pelo nome do material. O material é caracterizado não só pelo nome mas também por um identificador único e pela quantidade disponível. Deve ser permitido: adicionar novo material, remover material (porque ficou obsoleto), alterar a quantidade (porque se atualizou o stock – efetuou uma compra ou o material foi usado numa cirurgia). Devem ainda ser permitidas listagens várias.
- O hospital deve manter um registo das cirurgias já realizadas. Atualize o seu programa de modo a guardar esses registos históricos numa **tabela de dispersão**. Deve ser permitido a inserção de registos na tabela de dispersão, remoção de registos da tabela de dispersão (porque excedeu determinada data limite) ou alteração de dados desses registos. Use como chave uma função do nome do paciente e da cirurgia realizada
- Considere que a situação de cada paciente tem uma prioridade associada e que as cirurgias são efetuadas de acordo com essa prioridade. Todas as cirurgias planeadas devem ser guardadas numa única **fila de prioridade**, onde as situações são ordenadas por uma função que combina a urgência da doença com a idade do paciente. Para calcular a prioridade use a seguinte fórmula: $pri = C1 * cirurgia + C2 * 1/idade$

C1 e C2 são coeficientes que pode fixar como constantes do programa. Para os valores do termo *cirurgia* construa uma tabela de prioridades (entre 0 e 1) para cada tipo de cirurgia.

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.

Trabalho 9 – Proteção Civil (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Guarde a informação relativa aos postos de socorro numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pela designação do posto de socorro. Deve ser permitido: adicionar novos postos de socorro, remover postos de socorro (porque foram extintos), alterar informação do posto de socorro. Devem ainda ser permitidas listagens várias.
- Considere que se tem de guardar um histórico das ocorrências dos acidentes resolvidos. Para tal deve utilizar uma **tabela de dispersão**. Utilize o identificador do acidente e o local para implementar a função de inserção na tabela de dispersão (*hash function*). Deve ser permitido a inserção de acidentes na tabela de dispersão, remoção de acidentes da tabela de dispersão (porque já não excederam um determinado limite de tempo) ou alteração de dados desses candidatos (eventuais correção de dados).
- Considere que os acidentes reportados ao centro de Proteção Civil devem ser atendidos não por ordem de chegada, mas sim por prioridade. Implemente, com recurso a uma **fila de prioridade**, o atendimento priorizado dos acidentes. Estabeleça uma função razoavelmente verosímil para atribuição de prioridades (ex. Número de sinistrados, número de viaturas envolvidas, ou área do incêndio etc). Deve ser possível também remover um acidente, quando se verifica que foi um falso alarme.

Nota: O trabalho deve ilustrar a realização das operações básicas CRUD (*Create, Read, Update, Delete*) sobre as estruturas de dados: árvore binária de pesquisa, fila de prioridade e tabela de dispersão.