

## Trabalho 1 - Gestão de uma Oficina Mecânica (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Quando é necessário que o cliente deixe a sua viatura na oficina por um longo período de tempo, a oficina disponibiliza um serviço de ajuda ao aluguer de uma viatura de substituição. Considere uma **fila de prioridade** que guarda informação sobre empresas de aluguer de viaturas. Uma empresa de aluguer de viaturas é caracterizada pela sua denominação, localização e viaturas disponíveis. O sistema deve encontrar a viatura que serve os requisitos do cliente (tipo viatura, número de lugares/capacidade de carga, tipo combustível, ...) que se encontra disponível na empresa de aluguer mais próxima. Deve ser permitido: efetuar alugueres de viaturas (diminuindo assim o número de viaturas disponíveis na empresa de aluguer) ; adicionar novas empresas de aluguer e respetivas viaturas ; remover empresas de aluguer (porque a empresa deixou de existir).
- 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).
- Os produtos/peças usados nos serviços que a oficina oferece são armazenados numa **árvore binária de pesquisa**, sendo a ordenação efetuada por designação do produto. Um produto é descrito por um código, designação, stock e loja onde se adquire. Deve ser permitido: adicionar novos produtos, remover produtos (porque já não se usam), alterar o stock (porque se comprou ou usou determinado produto). Devem ainda ser permitidas listagens várias.

*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 Salas de uma Escola (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Na leccionação de aulas, pode ser necessário requisitar docentes externos à escola. Considere uma **fila de prioridade** que guarda informação sobre docentes externos à escola. Um docente externo é caracterizado pelo nome, contacto, categoria e disciplinas da sua competência. Para a leccionação de determinada aula, o sistema deve encontrar o docente externo mais apto (de categoria mais elevada) que possui essa disciplina no seu leque de competências. Deve ser permitido: adicionar um novo docente externo e respectivas competências; alterar o contacto ou as competências de um docente externo; remover docentes externos (deixam de estar disponíveis).
- Nos eventos de *workshop* participam oradores convidados especialistas. Para contacto futuro, ou simplesmente para arquivo de informação, os oradores convidados para *workshops* realizados na escola são guardados numa **tabela de dispersão**. A informação a guardar sobre um orador convidado (especialista) deve incluir nome, endereço electrónico, especialidade e data da última participação. Deve ser permitido: adicionar um especialista; remover um especialista (porque se pretende eliminar qualquer referência ao especialista); alterar dados de um especialista (endereço eletrónico, por exemplo).
- Considere que as diferentes escolas do concelho estão armazenadas numa árvore binária de pesquisa, sendo a ordenação efetuada por freguesia e dimensão (número de salas). Deve ser permitido: adicionar uma nova escola; remover uma escola (porque deixou de funcionar); alterar dados da escola (número de estudantes, por exemplo). Devem ainda ser permitidas listagens várias.

*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 uma Empresa Imobiliária (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- A empresa imobiliária fornece um serviço de ajuda ao cliente na sugestão de uma instituição de crédito quando o cliente o necessitar. Considere uma **fila de prioridade** que guarda informação sobre instituições de crédito com as quais a empresa tem contacto. Uma instituição de crédito é caracterizada pela sua denominação, número de clientes que aí possuem crédito à habitação e satisfação (sendo este valor o resultado de avaliações efetuadas pelos clientes). O sistema deve sugerir a instituição de crédito com maior valor de satisfação e, em caso de empate, com maior número de clientes com crédito à habitação. Deve ser permitido: avaliar uma instituição de crédito (alterando assim o valor da sua satisfação); adicionar um cliente a uma instituição de crédito ; adicionar novas instituições de crédito; remover instituições de crédito (a instituição de crédito deixou de existir).
- Os imóveis vendidos devem ser guardados numa **tabela de dispersão**. Deve ser permitido a inserção de imóveis na tabela de dispersão, remoção de imóveis (porque o imóvel deixou de existir, foi destruído) ou alteração de dados de um imóvel (a área do imóvel foi aumentada, por exemplo).
- A empresa imobiliária fornece um serviço de ajuda ao cliente na sugestão de lojas de decoração. As lojas estão armazenadas numa **árvore binária de pesquisa**, sendo a ordenação efetuada por nome da loja. Além do nome, a loja é caracterizada também pela morada, contacto e um conjunto de tipos de produto que fornece (iluminação, artigos de cozinha, ...). Deve ser permitido: adicionar uma nova loja de decoração e respetivos produtos que comercializa; alterar os produtos que comercializa ou o contacto; remover uma loja de decoração (porque deixou de existir). Devem ainda ser permitidas listagens várias.

*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 Obras (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere que a Construtora pretende manter informação sobre empresas que nunca contratou mas que podem vir a ser sub-contratadas no futuro. Para o efeito deve criar uma nova classe **SubContratada** em que guarda a seguinte informação sobre cada empresa: nome; tipo de trabalhos que realiza (domésticos ou infra-estruturas); número de empegados da empresa, volume médio anual de negócios (em milhares de euros). Estas empresas, passíveis de serem sub-contratadas, devem ser armazenadas numa **árvore binária de pesquisa**, sendo a ordenação efetuada por nome da empresa. Deve ser permitido: adicionar uma nova empresa; alterar informação de uma empresa; remover uma empresa. Devem ainda ser permitidas listagens várias.
- O pagamento de cada trabalho a ser executado deve ser efectuado pela Tesouraria segundo um critério de prioridades. Sob o ponto de vista da tesouraria a prioridade associada é tanto maior quanto mais baixo for o custo do trabalho. Implemente uma **fila de prioridade** (a ser usada pela tesouraria) para guardar os trabalhos que se vão realizar e que respeite a prioridade baseada no seu custo. Deve ser permitido: submeter novos trabalhos; eliminar trabalhos das filas e atualizar a sua prioridade.
- A informação das empresas sub-contradas que já não prestam serviço à construtora deve ser guardada numa **tabela de dispersão** usando como chave o nome da empresa. Deve ser permitido a inserção de empresas na tabela de dispersão; remoção de empresas (porque o último serviço contratado é anterior ao ano AAAA – constante especificada no programa); alteração de dados da empresa (p.ex, contacto)

*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 - Gestão de um *Web Site Provider* (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere que o WSP tem um catálogo de *Web sites* para mostrar aos potenciais clientes. O catálogo tem informação sobre: protótipos de páginas (ex: pág. de um professor, pág. de uma Faculdade, pág. de uma livraria, pág. de empresa construtora, ...), custo da página, tempo de desenvolvimento, lista de tecnologias envolvidas. Os itens do catálogo devem ser armazenados numa **árvore binária de pesquisa**, sendo a ordenação efetuada por nome do protótipo. Deve ser permitido: adicionar um novo protótipo; alterar informação de um protótipo; remover um protótipo. Devem ainda ser permitidas listagens várias.
- Considere que os processos de cobrança dos *sites* desenvolvidos devem ser processado pela Tesouraria atendendo ao tipo de utilizador e a prioridades. Os pedidos de utilizadores particulares não têm prioridades associadas e devem ser guardados numa fila. Os pedidos efectuados por empresas têm uma prioridade associada. Esses pedidos são tanto mais prioritários quanto mais alto for o custo. Os pedidos de empresas devem ser guardados numa **fila de prioridade**. Deve ser permitido: submeter novas empresas; eliminar empresas das filas e atualizar a sua prioridade.
- Os utilizadores que terminaram o contrato com o WSP devem ser guardados numa **tabela de dispersão** usando como chave o nome do utilizador. Deve ser permitido a inserção de utilizadores na tabela de dispersão; remoção de utilizadores (porque o utilizador terminou o contrato há mais de N anos, sendo N uma constante da aplicação); alteração de dados de um utilizador (p.ex, contacto)

*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 *jobs* em “Grid Computing” (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere que o utilizador tem acesso a várias redes (*grids*) temáticas onde pode correr os seus *jobs* conforme o domínio da aplicação do *job*. Considere que existem *grids* para problemas de física, medicina, genética, engenharia civil, etc. Implemente uma nova classe *Grid* para guardar informação relativa a uma *grid* que inclui: o nome da *grid*; o tema, o número de máquinas disponíveis; a capacidade de armazenamento em disco; a capacidade de RAM da máquina com a maior capacidade de RAM (em GB). As *grids* devem ser armazenadas numa **árvore binária de pesquisa**, sendo a ordenação efetuada por nome da *grid*. Deve ser permitido: adicionar uma nova *grid*; alterar informação de uma *grid*; remover uma *grid*. Devem ainda ser permitidas listagens várias.
- Cada *job* a ser executado deve ter uma prioridade associada. Um *job* é tanto mais prioritário quanto mais baixo for o valor da prioridade. Considere que tem máquinas de dois tipos: máquinas que ignoram a prioridade dos *jobs*; e máquinas que respeitam a prioridade. As máquinas que ignoram as prioridades guardam a lista de *jobs* a executar numa fila enquanto que as outras utilizam uma **fila de prioridade** para guardar os *jobs* a executar. Deve ser permitido: submeter novos *jobs*; eliminar *jobs* das filas e atualizar a sua prioridade.
- Em vez de um *job* ser imediatamente associado a uma máquina (como acontecia no Trabalho 1) o *job* é colocado numa fila e quando é o próximo a ser executado é atribuído a uma máquina. Os utilizadores da aplicação que deixaram de utilizar os recursos informáticos devem ser armazenados numa **tabela de dispersão**, usando como chave o nome do utilizador. Deve ser permitido a inserção de utilizadores na tabela de dispersão, remoção de remoção (porque o utilizador deixou de utilizar os recursos há mais de N anos) e alteração dos seus dados.

*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 - Gestão de uma Cooperativa Vinícola (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Quando os produtores da cooperativa não têm transporte próprio, para levar os seus produtos aos armazéns da cooperativa, a cooperativa oferece este serviço, que é adjudicado a empresas transportadoras. Para isso, mantém uma carteira de transportadoras com quem trabalha numa **fila de prioridade**, que guarda informação sobre empresas de transporte. Uma empresa de transporte é caracterizada pela sua denominação, localização e viaturas disponíveis com uma capacidade de carga. O sistema deverá encontrar a viatura que satisfaz as necessidades do produtor, relativamente a carga que deseja transportar (note que poderão ser necessárias mais do que uma viatura), que se encontra na transportadora mais próxima. Deve ser possível requisitar transporte para uma determinada carga, adicionar novas transportadoras e respetivas viaturas, alterar a capacidade de carga de uma viatura, remover transportadoras, entre outras opções que julgue necessárias.
- A cooperativa mantém uma carteira de clientes inativos, constituídos por aqueles clientes que já não compram seus produtos há mais de um ano. Esta carteira de clientes inativos é guardada, para efeitos de publicidade e ofertas da cooperativa, numa **tabela de dispersão**. Deve ser permitido a inserção de clientes na tabela de dispersão, remoção de clientes da tabela de dispersão (porque já não pretendem receber informação a respeito de ofertas da cooperativa) ou alteração de dados desses clientes (contacto, por exemplo).
- Os vinhos comercializados pela cooperativa são guardados numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pela designação do vinho. Cada vinho será identificado com um código, terá uma designação, e outras informações, como ano da colheita, quantidade em stock, acidez (maduro ou verde), tipo (branco, tinto), graduação (em %), castas que integram a sua constituição, região. Deve ser permitido: adicionar vinhos, remover vinhos (porque já não estão disponíveis em stock), e alterar preço. Devem ainda ser permitidas listagens várias.

*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 uma Agência de Manutenção Doméstica (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Imagine agora que os serviços específicos de lavar e engomar já não são realizados ao domicílio, tendo a empresa decidido realizá-los na própria sede, depois de adquirir equipamentos industriais de lavagem e secagem de roupa. Algumas colaboradoras são, portanto, escaladas para engomar, completando assim estes serviços. As roupas de cada cliente são processadas separadamente, em cada um dos passos (lavar, secar, engomar). As máquinas de lavar e secar, têm uma capacidade, em quilos, e um tempo de processamento para meia carga e carga total. Por outro lado, as colaboradoras que passam a ferro gastam, cada uma, um tempo médio para passar um determinado número de peças. Todos esses recursos são postos a servir uma única **fila de prioridade**, onde os pedidos são ordenados por peso das peças incluídas no pedido e, em caso de empate, por antiguidade do cliente. Antes dos pedidos serem processados, é possível ao cliente adicionar mais itens ao seu pedido. Os pedidos podem ser eliminados (o cliente cancelou o pedido).
- São considerados clientes inativos da empresa, aqueles que já não solicitam qualquer tipo de manutenção doméstica há mais de um ano. Entretanto, para efeitos de publicidade e promoções, deseja-se guardá-los 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).
- Para o serviço de lavar, secar e passar, as peças de vestuário são guardadas numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pela designação da peça de vestuário. Uma peça de vestuário é descrita por um código, designação, peso médio (em Kg), tipos de cuidados específicos (relativamente a poder lavar a seco, secar em máquina, e passar a ferro) e número de unidades a serem processadas, à medida que os pedidos chegam. Deve ser permitido: adicionar novas peças de vestuário, remover peças (porque já não são aceites), alterar o número de unidades em processamento (porque se concluiu serviços ou por novos serviços pedidos). Devem ainda ser permitidas listagens várias.

*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 - Gestão de uma Agência de Empregos (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Imagine agora que a agência de empregos realiza uma seriação dos candidatos antes de sugerirlos às empresas numa **fila de prioridade**, onde o critério de ordenação é uma combinação (somatório) da valoração que a empresa solicitante indica para as características dos candidatos, como menor distância da morada ao posto de trabalho, habilitações, experiência prévia e disponibilidade (caso o candidato esteja empregado e necessite de tempo para rescindir contrato). Imagine também que, durante o período em que a vaga está disponível e se realiza o processo de seriação, candidatos podem atualizar o seu CV, incluindo habilitações, alterando morada, disponibilidade. Deve ser possível também remover um candidato, quando este deixa de estar interessado em oportunidades de emprego.
- Na agência de empregos, são considerados candidatos inativos os que já foram efetivamente colocados num posto de trabalho. Entretanto, para efeitos de publicidade de novos empregos ou oportunidades, deseja-se guardá-los numa **tabela de dispersão**. Deve ser permitido a inserção de candidatos na tabela de dispersão, remoção de candidatos da tabela de dispersão (porque já não pretendem receber informação a respeito de oferta de empregos) ou alteração de dados desses candidatos (contacto e emprego atual, por exemplo).
- A agência de empregos guarda todas as ofertas de trabalho apresentadas pelas empresas numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pela designação do posto de trabalho. Cada posto de trabalho oferecido será identificado com um código, terá uma designação, e outras informações, como número de vagas para aquela oferta, período em que a oferta está disponível para candidaturas (data de início e data de fim), e outras características do posto, como jornada de trabalho, salário, entre outras. Deve ser permitido: adicionar novas ofertas de trabalho, remover ofertas (porque o período de candidatura já expirou), alterar salário, e alterar o número de vagas para aquela oferta (porque entretanto já foi admitido alguém). Devem ainda ser permitidas listagens várias.

*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.