

## Trabalho 1 - Gestão de uma Biblioteca (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere que os livros não sujeitos a empréstimo são agora guardados numa **árvore binária de pesquisa**. Complete a informação do livro com o ano de edição do livro e implemente a ordenação da árvore por ano de edição (os mais antigos em primeiro lugar), pelo título e pelos nomes dos autores. Deve ser permitido: adicionar novos livros à biblioteca, remover livros (porque se extraviaram ou foram doados a outras bibliotecas, por exemplo), alterar informações do livro. Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Para efeitos de promoção da leitura, a biblioteca passou a guardar numa **tabela de dispersão** todos os seus leitores inativos, isto é, que já não realizam empréstimos de livros há mais de 12 meses. Para efeitos de envio de newsletters e convites para eventos a realizar na biblioteca, interessa manter os contatos dos leitores atualizados (email, morada e telefone). Deve ser permitido a adição de leitores inativos, remover registos dos leitores (assim que realizem um empréstimo), ou alterar os seus dados, como morada, telefone, etc. Deve ainda ser possível listar os leitores inativos.
- Para melhorar a gestão dos empréstimos de livros, a biblioteca decidiu implementar uma fila de espera quando não há exemplares disponíveis de um determinado livro. Assim os pedidos de empréstimo em espera são guardados numa **fila de prioridade**, em que os principais critérios de prioridade são a data do pedido e o tipo de leitor (estudante, criança com idade inferior ou igual a 12 anos e adulto). Deve ser permitido: adicionar novos pedidos de empréstimo à fila de espera, cancelar empréstimos quando o leitor deixa de estar interessado no livro ou assim que haja um exemplar disponível para empréstimo, ou alterar informações do pedido.

*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 AppStore (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- De modo a apresentar o top 10 de apps aos clientes, decide-se manter a informação das apps numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada por classificação da app, preço e categoria. Deve ser permitido adicionar apps, remover apps, e alterar a informação das apps. Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore. Nota: na árvore de pesquisa não deve colocar diretamente as apps existentes, mas sim referências para as apps já presentes nas estruturas de dados consideradas na 1ª parte do trabalho.
- Considere que as apps que o developer decidiu deixar de vender devem ser guardadas numa **tabela de dispersão**. Deve ser permitido, na tabela de dispersão: a inserção de apps que o developer decidiu deixar de vender; a remoção de apps, caso o developer decida publicá-las novamente para venda; e a alteração das características das apps que estão na tabela. Deve ainda ser possível listar as apps presentes na tabela de dispersão.
- Considere que a appstore se tornou bastante popular e que um crescente número de developers a procura para vender as suas apps. A publicação de apps é agora sujeita a um processo de validação antes da sua publicação na appstore. Para isso mantenha, mantenha as apps submetidas para validação numa **fila de prioridade**. A ordenação das apps na fila deve obedecer aos seguintes critérios: data de submissão, preço da app e nome da app. Deve ser possível inserir, remover e alterar as apps 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 3 - Gestão de um Clube Desportivo (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Para a modalidade de futebol, os sócios podem fazer a reserva de lugares no início de cada época. A informação sobre a reserva de lugares de cada sócio é guardada numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada por tipo de lugar e nome de sócio. A informação a manter sobre a reserva de lugares inclui tipo, custo anual (segundo a tabela abaixo) e data de pagamento. Deve ser permitido adicionar reservas de lugares; remover reservas de lugares (porque o sócio demonstrou interesse mas não chegou a pagar ou porque foram realizadas já há bastante tempo e não interessa mantê-las); e, alterar reservas de lugares (por exemplo, o sócio pretende mudar de tipo de lugar). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.

Tipo de Lugar	Custo Anual
Camarote	500 €
Cativo	100€
Anual	60€

- Os sócios com quotas em atraso devem ser guardados numa **tabela de dispersão**. Deve ser permitida a inserção de sócios com quotas em atraso na tabela de dispersão, remover registos da tabela (quando o sócio paga a sua dívida) ou alterar os dados do sócio, para efeitos de contacto, por exemplo, atualização de morada, e para atualizar o valor em dívida. Deve ainda ser possível listar os sócios com quotas em atraso.
- De forma a reduzir as despesas com pessoal externo, que se centram essencialmente em serviços específicos de manutenção, o clube decidiu adjudicar os serviços de manutenção a empresas especializadas. A informação sobre esses fornecedores de serviços passa a ser guardada numa **fila de prioridade**. Um fornecedor de serviços de manutenção é caracterizado pela sua denominação, NIF, localização e serviços específicos que presta. O sistema deve encontrar o fornecedor mais próximo que possui o serviço que o clube pretende contratar. Deve ser permitido: adicionar novos fornecedores e serviços associados; remover fornecedores (quando deixam de existir); editar fornecedores (por exemplo, quando se dá uma mudança de instalações e edição de serviços que presta) .

*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 – Reserva de bilhetes de avião (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Os aviões estão sujeitos a serviços de manutenção e limpeza (entre outros), cuja data de realização é determinada antecipadamente. A informação sobre todos os serviços de manutenção e limpeza dos aviões é guardada numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada por data e, para a mesma data, por matrícula do avião. Um serviço é caracterizado por tipo de serviço (manutenção/limpeza), data e estado (realizado ou não). Deve ser permitido: adicionar serviços; remover serviços (porque já não irão ser realizados, ou foram realizados já há bastante tempo e não interessa mantê-los); alterar a informação de um serviço (a data de realização do serviço foi alterada, ou o seu estado). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Para a definição completa de um voo, é necessário alocar tripulantes (pilotos, hospedeiras). Os tripulantes da companhia aérea são guardados numa **tabela de dispersão**. Deve ser permitido: adicionar tripulantes na tabela de dispersão; remover tripulantes (porque este se aposentou ou deixou de trabalhar na companhia aérea); alterar os dados do tripulante (atualização de contacto ou morada, por exemplo). Deve ainda ser possível listar os tripulantes da companhia aérea.
- Para auxílio aos passageiros, a companhia aérea mantém informação sobre os locais de transporte terrestre junto de cada aeroporto de destino das suas rotas. Para cada aeroporto, a informação sobre os locais de transporte terrestre (metro, comboio, autocarro) está guardada numa **fila de prioridade**. Um local de transporte terrestre é caracterizado pelo tipo de transporte (metro, autocarro, comboio), distância ao aeroporto, horários, .... Considere que a fila de prioridade está ordenada por **distância** do local de transporte ao aeroporto. O sistema deve encontrar o local de transporte mais próximo que possua o tipo de transporte terrestre (metro, autocarro, comboio) que o passageiro pretende. Deve ser permitido: adicionar novos locais de transporte; remover locais de transporte (porque são desativados); alterar locais de transporte (por exemplo, quando se altera o horário) .

*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 Hoteleira (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- O hotel dispõe de um SPA que disponibiliza diversos tratamentos para os seus clientes mediante marcação prévia. A informação sobre os tratamentos disponibilizados no SPA do hotel é guardada numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada por preço e, para o mesmo preço, por duração do tratamento. Um tratamento é caracterizado por designação, preço, duração e uma lista de clientes com marcação efetuada. Deve ser permitido: adicionar tratamentos; remover tratamentos (porque deixaram de ser disponibilizados pelo SPA); alterar a informação de um tratamento (alteração do preço, ou a marcação por parte de um cliente). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Os clientes mais antigos (em que a última reserva no hotel foi efetuada há mais de 1 ano) são guardados numa **tabela de dispersão**. Deve ser permitido: adicionar clientes antigos na tabela de dispersão; remover clientes antigos (porque se pretende mesmo eliminar qualquer referência ao cliente); alterar os dados do cliente (alteração de contacto ou morada, por exemplo). Deve ainda ser possível listar os clientes antigos presentes na tabela de dispersão.
- Para auxílio aos clientes, o hotel dispõe de informação sobre os restaurantes nas redondezas. A informação sobre um restaurante inclui distância deste ao hotel, categoria (função preço/qualidade), tipo de comida (tradicional, italiano, japonês, ...). Assim, o hotel guarda a informação sobre os restaurantes numa **fila de prioridade** ordenada por distância. O sistema deve encontrar o restaurante mais próximo que satisfaça os requisitos do cliente em termos de qualidade e tipo de comida. Deve ser permitido: adicionar novos restaurantes; remover restaurantes (porque fecharam); alterar informação do restaurante (por exemplo, alterar a categoria) .

*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 informação da Volta a Portugal (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Suponha que a organização da Volta é realizada por elementos da Federação Nacional de Ciclismo, de forma rotativa. Isto é, para a organização da Volta num determinado ano, são escolhidos, de entre os membros da Federação Nacional de Ciclismo, os 3 membros que menos organizações realizaram. A informação sobre os membros da Federação Nacional de Ciclismo é guardada numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada por nº de organizações já realizadas e, para o mesmo nº de organizações realizadas, por ordem alfabética de nome. Deve ser permitido: adicionar membros; remover membros (porque deixaram de pertencer à Federação); alterar a informação de um membro (alteração do contacto, ou nº de organizações). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Os patrocinadores (atuais e antigos) da volta são guardados numa **tabela de dispersão**. A informação sobre o patrocinador deve incluir (entre outros itens): nome, contacto, lista de equipas patrocinadas e respetivos anos. Deve ser permitido: adicionar patrocinadores; remover patrocinadores (porque se pretende mesmo eliminar qualquer referência ao patrocinador); alterar os dados do patrocinador (alteração de contacto ou da lista de equipas patrocinadas). Deve ainda ser possível listar os patrocinadores presentes na tabela de dispersão.
- No caso de ocorrer um acidente com qualquer ciclista, e se tal se justificar, este deve ser transportado para o estabelecimento de saúde mais próximo. Para facilitar a pesquisa do estabelecimento de saúde apropriado, é usada uma **fila de prioridade**. A fila de prioridade guarda a informação dos estabelecimentos de saúde próximos de um local e existe uma fila de prioridade associada a cada um dos locais onde passa a volta (considere apenas os locais associados ao início e fim das etapas). Os estabelecimentos de saúde são caracterizados por nome, tipo (hospital/centro de saúde) e distância. A fila de prioridade está ordenada por distância do estabelecimento de saúde ao local respetivo. O sistema deve encontrar, para um determinado local da volta, o estabelecimento de saúde mais próximo capaz de tratar o tipo de ferimento do ciclista (hospital ou centro de saúde). Deve ser permitido: adicionar novos estabelecimentos de saúde; remover estabelecimentos de saúde (porque foram desativados); alterar informação do estabelecimento de saúde (por exemplo, alterar o tipo).

*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 – Boleias Inteligentes (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Suponha que as empresas que oferecem serviços de transporte coletivo guardam o registo dos seus motoristas numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pela menor distância da morada do motorista à garagem da empresa e, quando houver motoristas com residência à mesma distância, por ordem alfabética do nome do motorista. Deve ser permitido: adicionar motoristas; remover motoristas (porque deixaram de trabalhar com a empresa); alterar as informações de um motorista (como a indicação de uma nova morada). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Considere agora que as empresas, para efeitos de marketing, decidem guardar os seus clientes que há muito não utilizam seus serviços de transportes, enviando-lhes publicidade e promoções. Estes clientes antigos passam a ser guardados numa **tabela de dispersão**, sempre que deixarem de utilizar serviços de transportes por um longo período de tempo (predefinido pela empresa, e.g. 12 meses). A informação sobre os clientes antigos deve incluir (entre outros itens): nome, contacto, morada, registo das viagens realizadas com a empresa e respectivos destinos. À tabela de dispersão deve ser permitido: adicionar clientes; remover clientes (porque passaram a utilizar novamente os seus serviços de transportes); alterar os dados do cliente (alteração do contacto ou morada). Deve ainda ser possível listar os antigos clientes presentes na tabela de dispersão.
- Considere ainda que os particulares que oferecem boleia utilizam uma **fila de prioridade** para os ajudar a encontrar os melhores passageiros dentre os candidatos às suas ofertas. Aos candidatos conhecidos, ou seja, àqueles que já partilharam viagens, é dada prioridade na escolha. No caso de haver candidatos com o mesmo histórico de partilha de viagens, é dada preferência àqueles com morada mais próxima da morada de que está a oferecer boleia. O sistema deve permitir identificar os melhores candidatos para o número de vagas disponíveis no veículo associado à oferta. Deve ser permitido adicionar novos candidatos, remover candidatos (porque entretanto já não precisam da boleia), alterar morada dos candidatos 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 – “Bike sharing” (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Suponha que a cooperativa mantém o registo dos proprietários das bicicletas numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pelo número de bicicletas disponibilizadas pelo proprietário, sendo o total de utilizações das bicicletas o critério de desempate para proprietários com o mesmo número de bicicletas disponíveis. Deve ser permitido: adicionar proprietários; aumentar ou diminuir o número de bicicletas disponíveis por proprietário; remover proprietários (porque deixaram de fornecer bicicletas para a cooperativa); alterar as informações de um proprietário (como a indicação de uma nova morada ou disponibilização de mais uma bicicleta). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Considere agora que a cooperativa, para efeitos de marketing, decide guardar os seus clientes que há muito não utilizam as suas bicicletas, enviando-lhes publicidade e promoções. Estes clientes antigos passam a ser guardados numa **tabela de dispersão**, sempre que deixarem de alugar bicicletas por um longo período de tempo (predefinido pela cooperativa, e.g. 12 meses). A informação sobre os clientes antigos deve incluir (entre outros itens): nome, contacto, morada, e registos das suas utilizações, com respectivos pontos de levantamento e entrega da bicicleta. À tabela de dispersão deve ser permitido: adicionar clientes; remover clientes (porque passaram a utilizar novamente as bicicletas da cooperativa); alterar os dados do cliente (alteração do contacto ou morada). Deve ainda ser possível listar os antigos clientes presentes na tabela de dispersão.
- Considere que a cooperativa tem um plano de manutenção das bicicletas, com a finalidade de mantê-las sempre em perfeitas condições de uso: todo mês retira de circulação um certo número de bicicletas, que são recolhidas ao armazém da cooperativa para serem submetidas à manutenção. A selecção é realizada segundo o critério de maior utilização. A cooperativa mantém o log de utilização das suas bicicletas (tempo de uso) numa **fila de prioridade**, onde as mais utilizadas são mantidas na frente da fila, e recolhidas mais cedo à manutenção. O sistema deve permitir identificar as bicicletas mais utilizadas, sempre que o processo de manutenção se repetir. Sempre que uma bicicleta retorna da manutenção, passará a ter utilização zero, passando ao fim da lista. A cada nova utilização, as bicicletas poderão ter sua ordem na fila alternada. Nota: na árvore de pesquisa não deve colocar diretamente as bicicletas existentes, mas sim referências para as bicicletas já presentes nas estruturas de dados consideradas na 1ª parte do trabalho

*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 Projetos (Parte 2)

Complemente o sistema já implementado com as seguintes funcionalidades:

- Considere que a empresa mantém uma carteira dos seus clientes numa **árvore binária de pesquisa**, sendo a sua ordenação efetuada pelo montante acumulado pago pelos projetos que o cliente já contratou à empresa; clientes com o mesmo valor pago acumulado são diferenciados pelo número de projetos já contratados. Apenas se considera projetos já concluídos, e os maiores valores acumulados e número de projetos concluídos são considerados primeiro. Deve ser permitido: adicionar clientes à carteira; atualizar o número de projetos e o valor acumulado pago pelo cliente; remover clientes (porque deixaram de trabalhar com a empresa); alterar as informações de um cliente (como os seus dados fiscais). Devem ainda ser permitidas listagens várias tirando partido da ordenação da árvore.
- Considere agora que a empresa deseja manter um registo dos seus antigos colaboradores numa **tabela de dispersão**, a fim de poder readmiti-los caso precisem de esforço adicional para novos projetos. A informação sobre os colaboradores antigos deve incluir (entre outros itens): nome, contacto, morada, lista de projetos em que esteve envolvido enquanto fazia parte do quadro da empresa e seu respetivo papel no projeto. À tabela de dispersão deve ser permitido: adicionar antigos colaboradores; remover antigos colaboradores (porque foram readmitidos para participarem de um projeto); alterar os dados do colaborador (alteração do contacto ou morada). Deve ainda ser possível listar os antigos colaboradores presentes na tabela de dispersão.
- Quando a empresa não tem pessoal suficiente para dar resposta ao número de projetos que está a executar, recruta colaboradores temporários, abrindo concursos para recrutamento de pessoal. Para facilitar a escolha dos candidatos, a empresa mantém as candidaturas numa **fila de prioridade**, ordenadas pelo tempo de experiência dos candidatos, pelo número de diferentes competências apresentadas no seu CV, e por ter sido ou não colaborador da empresa no passado. O sistema deve ser capaz de identificar os melhores candidatos para o número de vagas oferecidas. Depois de admitidos, os candidatos escolhidos passam a fazer parte do quadro de colaboradores da empresa (até que se desliguem da empresa, por alguma razão).

*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.