

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

Desenvolva uma aplicação em C++ para gestão de uma oficina mecânica. O sistema deve conter informação sobre veículos e serviços efectuados, clientes e funcionários.

A oficina aceita efectuar serviços sobre diferentes tipos de veículos: automóveis, motorizadas, camiões, autocarros,....

Todos os veículos possuem um funcionário responsável que responde perante o cliente sobre os serviços efectuados no seu veículo. Procure manter uma distribuição equilibrada de veículos a funcionários responsáveis.

A oficina oferece serviços:

- *standard*: revisão, mudança óleo, limpeza,... A descrição, preço e duração possuem valores pré-definidos. Pode considerar outros atributos, se necessário.
- *não standard*. A descrição, preço e duração são valores a definir quando da criação do serviço.

Alguns requisitos obrigatórios são (pode e deve incluir outros que considere relevantes):

- Manutenção de clientes
- Manutenção dos veículos
- Manutenção de serviços *standard*
- Manutenção de funcionários
- Associar funcionários e veículos (um funcionário pode ser responsável por mais que um veículo)
- Associar clientes e veículos (um cliente pode possuir mais que um veículo)
- Associar veículos e serviços (um veículo pode ter sido sujeito a mais que um serviço)
- ... outras associações que julgue necessárias

A aplicação deve permitir registar e gerir toda a informação relativa a quais serviços foram efectuados pela oficina e quando.

Considere o conjunto de algumas das entidades implementadas: Cliente, Veiculo (entidade no topo da hierarquia de veiculos), Funcionario, entre outras. Usando *estruturas de dados lineares*, deve implementar:

- Manutenção do conjunto de clientes
- Manutenção do conjunto de veículos
- Manutenção do conjunto de funcionários
- Manutenção do conjunto de serviços *standard*
- Listagens várias (quer totais, quer parciais com critérios a definir pelo utilizador) de, por exemplo:
 - clientes
 - veículos
 - serviços efetuados, serviços efectuados por veículo, serviços efectuados por cliente
 - ...

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 2 - Gestão de Salas de uma Escola (Parte 1)

Desenvolva uma aplicação em C++ para gestão de salas de uma escola. As salas possuem uma identificação, capacidade, funcionário responsável e equipamento/material existente. Procure manter uma distribuição equilibrada de salas a funcionários responsáveis. As salas são usadas para diferentes eventos: aula, reunião, workshop (pode considerar outros eventos que considere interessante).

Todos os eventos incluem informação sobre início, duração e nº de participantes. Diferentes eventos são caracterizados por diferente informação: a aula é semanal e inclui o nome da disciplina, o docente que lecciona a aula,...; a reunião inclui o docente coordenador da reunião, participantes, ...; a workshop inclui designação e docente responsável.

O sistema deve conter e gerir informação sobre as salas existentes e sua ocupação. Deverá ainda incluir informação sobre docentes, funcionários e aulas.

Alguns requisitos obrigatórios (pode e deve incluir outros que considere relevantes):

- Manutenção de salas
- Manutenção de eventos
- Manutenção de docentes e funcionários
- Associar salas e eventos
- Associar salas e funcionários
- Associar aulas e docentes
- ... outras associações que julgue necessárias

Considere o conjunto de algumas das entidades implementadas: Sala, Evento (entidade no topo da hierarquia de tipos de eventos), Docente, Funcionario, entre outras. Usando *estruturas de dados lineares*, deve implementar:

- Manutenção do conjunto de salas
- Manutenção do conjunto de eventos para um período (1 semestre)
- Manutenção do conjunto de docentes
- Manutenção do conjunto de funcionários
- Listagens várias (quer totais, quer parciais com critérios a definir pelo utilizador) de, por exemplo:
 - salas (e respectiva ocupação/disponibilidade)
 - eventos
 - docentes (e respectivo horário)
 -

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 3 - Gestão de uma Empresa Imobiliária (Parte 1)

Desenvolva uma aplicação em C++ para gestão de uma empresa imobiliária. O sistema deve conter informação sobre os imóveis que comercializa (disponíveis, alugados ou vendidos), clientes e funcionários.

A empresa imobiliária comercializa imóveis de diferentes tipos: vivendas, apartamentos, lojas e armazéns. As vivendas e os apartamentos podem ser usados quer para habitação, quer para negócio. As lojas e os armazéns são usados para negócio. Os preços de venda ou aluguer de um imóvel dependem da área que possui, da localização, e de outros factores próprios de cada tipo de imóvel. Numa vivenda é ponderada a área exterior, o facto de ter piscina ou não, ...; num apartamento é ponderada a topologia, o facto de ter garagem ou não, ...; num armazém é ponderado o tipo de produto a guardar (congelação, inflamável,...), ...

O preço do aluguer depende ainda da duração prevista do aluguer.

Todos os imóveis possuem um funcionário responsável. Procure manter uma distribuição equilibrada de imóveis ainda disponíveis a funcionários responsáveis por estes. Um imóvel pode ter sido sujeito a vários alugueres e/ou vendas (mas sempre pelo mesmo funcionário responsável).

Alguns requisitos obrigatórios são (pode e deve incluir outros que considere relevantes):

- Manutenção de imóveis
- Manutenção de clientes
- Manutenção de funcionários
- Manutenção de contratos (aluguer, venda, ...; na parte 1 do trabalho considerar apenas os contratos de aluguer e venda)
- Associar funcionários e imóveis (um funcionário pode ser responsável por mais que um imóvel)
- Associar clientes, imóveis e contratos (um cliente pode possuir mais que um imóvel; um imóvel pode ter sido sujeito a mais que um contrato)
- ... outras associações que julgue necessárias

A aplicação deve permitir registar e gerir toda a informação relativa a quais imóveis foram alugados ou vendidos e quando.

Considere o conjunto de algumas das entidades implementadas: Imovel (entidade no topo da hierarquia de imóveis), Cliente, Funcionario, Contrato, entre outras. Usando *estruturas de dados lineares*, deve implementar:

- Manutenção do conjunto de imóveis
- Manutenção do conjunto de clientes
- Manutenção do conjunto de funcionários
- Listagens várias (quer totais, quer parciais com critérios a definir pelo utilizador) de, por exemplo:
 - imóveis (de acordo com tipo, último contrato, nºcontratos, funcionário, ...)
 - clientes
 - funcionários, ...

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 4 - Gestão de Obras (Parte 1)

Desenvolva uma aplicação em C++ para gestão das obras de uma empresa Construtora. O sistema deve conter informação sobre as obras da “carteira de encomendas” da Construtora, os diferentes tipos de trabalhos possíveis numa obra e, para cada obra, os trabalhos contratados.

Os trabalhos podem ser de dois grandes tipos: infra-estruturas ou domésticos. Os trabalhos de infra-estruturas podem ser: arruamentos ou saneamento. Os trabalhos domésticos podem ser: de trolha; de electricista; ou de carpinteiro. Todos os trabalhos têm uma duração em dias, um custo e o nome da empresa sub-contratada que os realiza. Todos os trabalhos de infra-estruturas têm a identificação da rua. Todos os trabalhos domésticos têm a identificação da habitação. Os trabalhos de arruamentos são caracterizados pela quantidade de asfalto que necessitam. Os trabalhos de saneamento e de trolha são caracterizados pela quantidade de betão. Os trabalhos de electricista são caracterizados pelo comprimento de cabo eléctrico. E, finalmente, os trabalhos de carpinteiro são caracterizados pela área de madeira.

A aplicação deve permitir registar toda a informação necessária para gerir as obras da Construtora. Alguns requisitos obrigatórios (pode e deve incluir outros que considere relevantes):

- Manutenção de obras
- Manutenção de trabalhos
- Manutenção de uma lista de empresas sub-contratadas
- Diferentes tipos de cálculos associados a cada obra e às empresas sub-contratadas.
- Associar trabalhos e obras
- Associar trabalhos e empresas sub-contratadas
- ... outras associações que julgue necessárias

Considere o conjunto de algumas das entidades implementadas: Construtora, Trabalho (entidade no topo da hierarquia de trabalhos realizados nas obras), entre outras. Usando *estruturas de dados lineares*, deve implementar:

- Manutenção da carteira de obras
- Manutenção da lista de empresas
- Listagem (quer totais, quer parciais com critérios a definir pelo utilizador) de:
 - obras
 - empresas
 - obras em que participa uma empresa
 - ...
- Cálculo da duração de cada obra
- Cálculo do custo de cada obra (total e por tipo de trabalhos)

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 5 - Gestão de um *Web Site Provider* (Parte 1)

Desenvolva uma aplicação em C++ para gestão de um *Web Site Provider*. Um *Web Site Provider* (WSP) permite a utilizadores particulares ou empresas usarem os recursos da WSP para alojar e desenvolver sítios Web com URL próprio. O Gestor de um WSP é responsável por gerir um conjunto de sítios Web. Os sítios Web são de dois tipos: empresas ou particulares. Um sítio Web tem sempre um identificador e o número de páginas do sítio. Os sítios Web das empresas têm uma lista de utilizadores que podem fazer a sua gestão, o custo por página para empresas, e a lista de tecnologia que podem usar (HTML, PHP, ...). Os sítios Web de particulares têm o nome do gestor do sítio, um limite no número de páginas, o custo por página para sítios particulares e a tecnologia usada (só uma). O sistema deve ainda manter uma lista de utilizadores gestores dos sítios empresariais e individuais.

Alguns requisitos obrigatórios (pode e deve incluir outros que considere relevantes):

- Manutenção de sítios Web
- Manutenção de utilizadores
- Associar gestores e WSP
- ... outras associações que julgue necessárias

Considere o conjunto de algumas das entidades implementadas: GestorWSP, WebSite (entidade no topo da hierarquia de tipos de sítios Web), Utilizador, entre outras. Usando *estruturas de dados lineares*, deve implementar:

- Manutenção do conjunto de sítios Web
- Manutenção da lista de utilizadores
- Listagem (quer totais, quer parciais com critérios a definir pelo utilizador) de:
 - tipo de sítio
 - tecnologias usadas
 - ...
- Cálculo do custo total e por sítio Web

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 6 - Gestão de *jobs* em “Grid Computing” (Parte 1)

Desenvolva uma aplicação em C++ para gestão de *jobs* de utilizadores num ambiente de Grid Computing.

O sistema a implementar deve permitir gerir um conjunto de utilizadores que submetem *jobs* ao sistema Grid, fazer a atribuição de *jobs* às máquinas disponíveis e gerir o parque de máquinas.

Os utilizadores podem ser do tipo académicos ou empresas. Todos os utilizadores têm uma identificação (nome) e uma lista de *jobs* que pretendem executar. Os utilizadores académicos têm um contador de *jobs* já executados. Os utilizadores do tipo empresa têm uma verba atribuída e só podem executar *jobs* se tiverem verba disponível (assuma que o preço de 1 *job* é proporcional ao tamanho de RAM requerido para o executar).

Um *job* é caracterizado por um identificador, uma prioridade, uma quantidade de RAM e disco necessários, uma lista de software (nomes reconhecidos pelo sistema) que deve estar instalado na máquina onde vai correr. Uma máquina é caracterizada pela quantidade de RAM e disco disponível, pelo software instalado, pelo número máximo de *jobs* que aceita e pelo número e lista de *jobs* em execução.

Considere o conjunto de algumas das entidades implementadas: GestorGrid, Utilizador (entidade no topo da hierarquia de tipos de utilizadores), Máquina, Job, entre outras. Usando **estruturas de dados lineares**, deve implementar:

- Manutenção de utilizadores
- Manutenção de máquinas
- Manutenção de *jobs*
- Listagem (quer totais, quer parciais com critérios a definir pelo utilizador) de:
 - utilizadores
 - máquinas
 - *jobs* de uma máquina ou utilizador
 - ...
- Atribuição e remoção de *jobs* a máquinas

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 7 - Gestão de uma Cooperativa Vinícola (Parte 1)

Desenvolva uma aplicação em C++ para a gestão de uma cooperativa vinícola, constituída por produtores de vinhos em Portugal. A cooperativa é especializada em diversas bebidas derivadas de uva, incluindo vinhos verdes, maduros (brancos, tintos, rosês), vinho do Porto, espumantes, entre outros, assim como derivados de vinhos, como vinagres, etc. Os produtores produzem uvas em diversas regiões de Portugal, e especializam-se em determinadas castas. Os vinhos são todos produzidos a partir de uma casta ou combinação de várias castas. O preço de um vinho depende da casta ou castas que contém mas também de fatores próprios do seu tipo de vinho (por exemplo, o preço do vinho do Porto depende do ano da colheita e tempo de armazenamento em barril)

A cooperativa promove e comercializa os seus produtos a uma carteira comum de clientes, como supermercados, restaurantes, ou pequenos comércio, dentro e fora de Portugal..

A aplicação deve permitir registar toda a informação necessária à identificação dos produtores, dos clientes, dos produtos oferecidos pela cooperativa, assim como à sua comercialização...

O sistema a implementar deve registar toda a informação administrativa e contabilística da cooperativa, com os seguintes requisitos base (pode e deve incluir outros que considere relevantes):

- Manutenção de produtores, das castas que produz e dos seus respetivos produtos
- Manutenção de clientes
- Associar produtores e produtos
- Associar clientes e transações
- Associar produtores, clientes e histórico de transações
- ... outras associações que julgue necessárias

Considere o conjunto de algumas das entidades implementadas: Produtor, Produto, Transacao, Fatura, entre outras. Usando *estruturas de dados lineares*, deve efetuar:

- Manutenção do conjunto de produtores
- Manutenção do conjunto de clientes
- Manutenção do conjunto de produtos disponíveis
- Manutenção do histórico de transações
- Listagem (quer totais, quer parciais com critérios a definir pelo utilizador) de:
 - produtores
 - produtos
 - clientes
 - transações

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 8 - Gestão de uma Agência de Manutenção Doméstica (Parte 1)

Desenvolva uma aplicação em C++ para a gestão de uma Agência de Manutenção Doméstica. Muitas famílias, ao invés de manterem empregadas domésticas em tempo parcial ou total, optam por contratar serviços de manutenção a empresas especializadas. A agência em questão oferecerá serviços de manutenção doméstica, como passar, lavar, cozinhar, e arrumar, entre outros que possa identificar. Os serviços prestados no domicílio do cliente são pagos à hora, a depender dos itens contratados, do agregado familiar, e do tamanho da habitação. Adicionalmente, os itens lavar e passar podem ser contratados à parte, e pagos à unidade (ou peça de roupa). Os clientes, por sua vez, podem contratar serviços esporadicamente, ou contratar os serviços a longo prazo, sendo estes pagos mensalmente, de acordo com os serviços registados no mês.

A aplicação deve permitir registar toda a informação necessária à identificação dos serviços oferecidos pela agência, assim como à sua promoção (divulgação) entre os clientes e à sua comercialização, caso a empresa disponha de pessoal para o serviço. Os serviços no domicílio são sempre realizados por duplas de colaboradores da agência, que mantém um conjunto de colaboradores treinados para o efeito. É importante também realizar o agendamento dos serviços de forma a maximizar a utilidade dos colaboradores da agência (isto é, todos os colaboradores devem estar igualmente ocupados).

O sistema a implementar deve registar toda a informação administrativa e contabilística da agência, com os seguintes requisitos base (pode e deve incluir outros que considere relevantes):

- Manutenção de clientes e colaboradores da agência
- Manutenção do histórico dos serviços que a agência disponibiliza
- Associar clientes e serviços
- Associar equipas de manutenção (duplas de colaboradores) e clientes
- Associar clientes e histórico de serviços já prestados
- ...outras associações que julgue necessárias

Considere o conjunto de algumas das entidades implementadas: Cliente, EquipaManutencao, Servico, Fatura, entre outras. Usando *estruturas de dados lineares*, deve efetuar:

- Manutenção do conjunto de colaboradores da agência
- Manutenção do conjunto de clientes
- Manutenção do conjunto dos serviços disponíveis
- Manutenção do histórico de serviços realizados
- Listagem (quer totais, quer parciais com critérios a definir pelo utilizador) de:
 - serviços
 - colaboradores
 - clientes
 - serviços já efetuados

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático

Trabalho 9 - Gestão de uma Agência de Empregos (Parte 1)

Desenvolva uma aplicação em C++ para a gestão de uma Agência de Empregos, que é vocacionada à mediação da oferta de empregos, na área da engenharia, por empresas e candidatos.

A agência recebe com frequência candidaturas de interessados em diversos tipos de emprego. O candidato submete um conjunto de informações, quer pessoal (nome, morada, data nascimento,...), quer profissional (título académico, instituição que conferiu o grau, línguas, ..., outras informações próprias da área da engenharia em questão). Por exemplo, um engenheiro informático deve especificar as linguagens de programação que conhece, um engenheiro eletrotécnico a especialidade, etc.

As empresas com vagas disponíveis registam junto da agência as suas necessidades de pessoal, com a indicação do tipo de emprego (engenheiro informático, engenheiro eletrotécnico,...), carga horária exigida, e outros critérios próprios do tipo de emprego. As empresas devem também indicar qual é o funcionário da empresa responsável pela seleção dos candidatos. Deve identificar convenientemente o funcionário (nome, morada, cargo,).

A aplicação deve permitir registar toda a informação necessária à identificação das vagas disponíveis nas empresas, assim como dos candidatos e das suas habilitações. Quando uma nova vaga é registada na agência, esta notifica os candidatos em espera que possam ter interesse na vaga. No caso de haver vários candidatos, a empresa deverá realizar uma seriação, a fim de sugerir à empresa o melhor candidato. Caso não haja candidatos adequados, a vaga fica por preencher à espera de um candidato (quando surge um novo candidato devem ser verificadas todas as vagas). As vagas preenchidas devem ser guardadas para efeitos de análise estatística.

O sistema a implementar deve registar toda a informação administrativa da agência, com os seguintes requisitos base (pode e deve incluir outros que considere relevantes):

- Manutenção de empregos (ou competências)
- Manutenção de empresas e necessidades
- Manutenção de candidatos
- Associar candidatos e empregos
- Associar empresas e funcionários
- Associar empresas e candidatos e vagas preenchidas
- ...outras associações que julgue necessárias

Considere o conjunto de algumas das entidades implementadas: Emprego, Candidato, Empresa, Funcionario, entre outras. Usando *estruturas de dados lineares*, deve efetuar:

- Manutenção do conjunto de empresas e vagas que disponibilizam
- Manutenção do conjunto de candidatos
- Manutenção do conjunto de empregos
- Manutenção do histórico dos contratos celebrados com sucesso
- Listagem (quer totais, quer parciais com critérios a definir pelo utilizador) de:
 - Vagas disponíveis
 - Empresas (por vaga, por localidade,)
 - Candidatos (por competência, por idade, ...)
 - Contratos celebrados (por competência, por empresa, ...)
 - ...

Notas:

1. Por “manutenção de dados” entende-se as operações básicas CRUD (*Create, Read, Update, Delete*)
2. O trabalho deve respeitar o que está descrito em “Notas relativas à implementação” no texto relativo aos “Enunciados e instruções” da parte 1 do trabalho prático