

# Trabalho Prático de Técnicas de Programação (2003/2004)

## Regras para a implementação do trabalho prático:

1. O trabalho é realizado individualmente ou em grupos de dois alunos.
2. O trabalho é atribuído pelo docente segundo a política por este definida, não podendo o aluno entregar qualquer outro trabalho que não o que lhe tenha sido atribuído.
3. Os trabalhos encontram-se classificados de  $T_0$  a  $T_4$ . Ao aluno cabe apenas implementar um único  $T_n$  a indicar pelo docente (Ver adiante as regras de atribuição dos trabalhos).
4. O trabalho é para entregar impreterivelmente até ao dia 10 de Janeiro de 2004.
5. Os alunos devem entregar, até ao prazo estipulado, uma cópia dos ficheiros (disquete) e um relatório que aborde, entre outros, os seguintes pontos: descrição do problema; justificação das estruturas utilizadas; forma de modularização da implementação; descrição do funcionamento da aplicação; conclusões sobre os objectivos atingidos e não atingidos e, por último, uma listagem do programa devidamente comentada.
6. A avaliação do trabalho prático baseia-se na qualidade do relatório, na funcionalidade da aplicação desenvolvida (interface amigável e sucesso de utilização) e na defesa do próprio trabalho (obrigatória).
7. É da responsabilidade dos alunos garantirem que o trabalho funciona no sistema operativo e com o compilador utilizado nas aulas práticas.
8. A avaliação da época normal é atribuída pela média pesada das duas seguintes componentes:

|               |      |                  |
|---------------|------|------------------|
| 1ª Componente | 30 % | Trabalho prático |
| 2ª Componente | 70 % | Prova escrita    |

Os Docentes:

---

(13/10/2003)

## Atribuição do trabalho prático

O trabalho correspondente a cada grupo/aluno é determinado segundo o valor de  $x$  da tabela seguinte:

| Trabalho | $X$  |
|----------|------|
| T0       | 0, 5 |
| T1       | 1, 6 |
| T2       | 2, 7 |
| T3       | 3, 8 |
| T4       | 4, 9 |

Para os trabalhos realizados individualmente o valor de  $x$  é determinado pela função  $f1(...)$ :

```
int f1(int nmec){  
    return (nmec / 10) % 10;  
}
```

Onde  $nmec$  representa o número mecanográfico do aluno.

Para os trabalhos em grupo, o valor de  $x$  é determinado pela função  $f2(...)$ :

```
int f2(int nmec1, int nmec2){  
    int res;  
    res = (nmec1 / 10) % 10 + (nmec2 / 10) % 10;  
    while(res > 9) res = res -10;  
    return res;  
}
```

Onde  $nmec1$  e  $nmec2$  representam os números mecanográficos dos alunos.

NOTA: O aluno/grupo deve apenas implementar um único trabalho de entre T<sub>0</sub> a T<sub>4</sub>.

## T0 – Gestão de um Registo Civil

**Objectivo:** Desenvolver um software para apoio ao Registo Civil de Bragança

Pretende-se implementar um sistema de registo de bebés aos quais é atribuído um número de cédula e a seguinte informação: nome, sexo, data de nascimento, naturalidade, nome do pai e nome da mãe. Pretende-se também descobrir a sua árvore genealógica de ascendentes com base nas árvores do pai e da mãe. Cada nodo filho terá como descendentes o nodo pai e o nodo mãe.

### 1ª Parte – Módulo de gestão dos registos

Este módulo deverá permitir efectuar um novo registo; calcular uma árvore genealógica e armazená-la no registo respectivo; alterar a informação de um registo e manter actualizado o número de registos efectuados.

### 2ª Parte – Módulo de pesquisas

Este módulo deverá permitir procurar informação sobre uma determinada pessoa através do nº da cédula; calcular a sua idade; calcular os irmãos, pais, tios e primos de uma determinada pessoa, indicando os seus nomes e datas de nascimento.

## T1 – Gestão de abastecimentos numa rede de supermercados

**Objectivo:** Implementar um programa para gerir a rede de distribuição de produtos numa cadeia de supermercados

Considere que uma rede de supermercados pretende otimizar o processo de distribuição de produtos, mantendo um registo de todas as suas lojas e respectivas localizações. Sempre que um produto está pronto para ser distribuído determina-se o conjunto de lojas que necessitam de ser abastecidas e é dado, ao distribuidor, informação sobre o percurso a efectuar. No registo de cada loja deve constar o nome/cidade, a sua localização (coordenadas), data de abertura e uma lista de produtos em falta.

### 1ª Parte – Módulo de gestão de abastecimentos

Este módulo será responsável por registar uma nova loja (deverá ter como adjacente a loja mais próxima); receber informação das lojas sobre os produtos em falta; calcular (dado o nome do produto) o percurso a efectuar (caso o produto não esteja em falta em nenhuma das lojas ou a quantidade a distribuir não for suficiente não haverá distribuição); actualizar os dados relativos aos produtos em falta em cada uma das lojas no final da distribuição; e manter um historial de distribuições.

### 2ª Parte – Módulo de pesquisas

Este módulo deverá permitir listar todas as lojas pertencentes à rede; procurar informação sobre uma determinada loja; calcular o produto que está em falta num maior número de lojas.

## T2 – Gestão de um parque de estacionamento

**Objectivo:** Implementar um programa capaz de gerir a utilização diária de um parque de estacionamento gratuito.

Um parque de estacionamento tem  $n$  pisos com  $x$  lugares disponíveis. Quando todos os lugares de um piso estão ocupados os veículos devem ser encaminhados para os seguintes. Cada piso tem um nome,  $n^\circ$  de lugares e estado (aberto ou fechado). Quando todos os pisos estão preenchidos, os automóveis formam uma fila de espera única. A fila vai diminuindo à medida que os lugares vão vagando e é indicado o piso para o qual o condutor se deve dirigir. Considerando que existem dois tipos de veículos, classe A (ligeiros) e classe B (mais pesados), é política dos parques reservar o 1º piso para os mais pesados. Assim um veículo classe A só vai para o 1º piso quando todos os outros estão ocupados e um de classe B só não vai para o 1º piso quando este está ocupado.

### 1ª Parte – Módulo de inicialização

Este módulo será responsável por definir o parque de estacionamento ao qual o programa vai ser aplicado ( $n^\circ$  de pisos e  $n^\circ$  de lugares por piso); inicializar o estado dos lugares (todos desocupados no início do dia); criar uma fila de espera vazia.

### 2ª Parte – Módulo de gestão de lugares

Este módulo servirá para dar entrada de um automóvel na fila; indicar o piso disponível ou dizer para aguardar; registar as entradas e saídas actualizando os dados relativos a  $n^\circ$  de lugares ocupados em cada piso; abrir ou encerrar um piso do parque de estacionamento; manter um registo do número total de entradas.

### 3ª Parte – Módulo de pesquisas

Este módulo deverá permitir calcular o  $n^\circ$  de entradas diárias; calcular o número de viaturas em fila de espera; e consultar o estado e o  $n^\circ$  de lugares livres em cada piso.

## T3 – Gestão de uma casa de leilões

**Objectivo:** Implementar um pequeno programa para gerir uma casa de leilões on-line.

Uma casa de leilões tem, em cada momento, um conjunto de leilões a decorrer, agrupados por categoria de artigo a ser leiloado (electrodomésticos, automóveis, arte, etc.). Para cada leilão, interessa considerar a designação do artigo que está a ser leiloado, a data de início do leilão, a base de licitação inicial, a data de fim e uma lista com as licitações já obtidas, ordenadas por ordem cronológica inversa. O programa deverá também manter uma lista dos utilizadores registados na casa, identificados pelo endereço de e-mail. Apenas estes poderão efectuar licitações.

Quando um leilão é dado por terminado, ele é retirado da estrutura inicial e colocado numa outra, onde interessa guardar a categoria, a designação do artigo, a data de fecho do leilão, a licitação vencedora, o e-mail do vencedor e o número de licitações obtidas.

### 1ª Parte – Módulo de gestão das estruturas

Neste módulo será possível gerir a estrutura que suporta os utilizadores e os leilões, inserindo novos utilizadores, novos leilões e novas licitações (só para leilões cuja data de fim ainda não tenha passado). Note que a licitação tem que ser superior à última licitação efectuada para o artigo.

### 2ª Parte – Módulo de finalização e estatísticas

Neste módulo o gestor do sistema deverá poder finalizar um leilão (cuja data de fim deverá ser anterior à data do sistema). Nesta situação, o sistema deverá simular o envio de um e-mail ao vencedor.

Para os leilões terminados, interessa obter estatísticas, como por exemplo:

- o número de artigos leiloados em cada categoria;
- o número médio de licitações para cada categoria de artigos;
- o utilizador que mais leilões venceu.

## T4 – Gestão de uma empresa de construções civis XPTO

**Objectivo:** Implementar um pequeno programa para gerir uma empresa de construções civis em seu departamento de orçamentos.

A empresa XPTO de construções civis, possui um departamento de orçamentos, e este recebe pedidos de orçamentos para efectuarem obras. Quando um pedido chega o Eng.º responsável avalia e constata quais as empresas a subcontratar, consoante os serviços necessários, e a partir daí efectua o respectivo orçamento. O objectivo da aplicação a desenvolver, implica a existência de um repositório de informações relativas a cada empresa que são utilizadas para subcontratação, tendo cada uma delas uma lista de tipos de serviços e seus preços. No final o programa deverá ser capaz de permitir a criação e armazenamento de cada encomenda, sendo que cada encomenda é composta por vários tipos de serviços e seu respectivo preço unitário.

### 1ª Parte – Módulo de gestão das estruturas

Neste módulo será possível gerir a estrutura que suporta as empresas que podem ser subcontratadas pela empresa XPTO, com todas as suas identificações e os preços respectivos de cada tipo de serviço.

### 2ª Parte – Módulo de finalização e estatísticas

Neste módulo o gestor do sistema deverá poder gerar uma encomenda adicionando vários tipos de serviços de cada empresa possível de subcontratar e no final indicar qual o valor do orçamento.

Para os orçamentos efectuados o programa deverá permitir saber o seguinte:

- o número de vezes que um subcontratado foi incluído nos orçamentos;
- o valor médio por orçamento;
- o cliente que mais orçamentos pediu.