



Projeto Prático

1. Informação Geral

Considere a biblioteca de uma Escola Superior que possui um acervo de alguns milhares de livros, distribuídos fisicamente por diversas salas com estantes, segundo uma organização por áreas/temáticas, nomeadamente Programação, Design, Matemática, Hidráulica, etc., e que serve um vasto conjunto de leitores/requisitantes.

2. Descrição do Problema

Pretende-se desenvolver um programa para gerir o funcionamento da biblioteca mencionada acima, facilitando a gestão dos seus livros, da sua lista de requisitantes e ainda as requisições de livros por parte dos requisitantes.

Relativamente aos livros, é importante considerar a informação respeitante a ISBN, Título, Autor, Área e Ano de publicação. De notar que o ISBN é um identificador único para um determinado livro e existe sempre. A informação para alguns dos restantes campos pode estar omissa em alguns dos livros da Biblioteca. No que concerne aos requisitantes/leitores, cada um deles possui um identificador único, sendo ainda guardada informação relativamente ao nome, à data de nascimento e à naturalidade. Os dados relativos aos leitores são disponibilizados no ficheiro "*requisitantes.txt*", cujo formato é mostrado abaixo. O identificador do requisitante/leitor (*id_requisitante*) é atribuído pela Biblioteca no momento de registo de um novo leitor e é um identificador único. i.e., não podem existir 2 leitores com o mesmo identificador. Este *id_requisitante* é constituído por 9 algarismos, em que o último é de controlo. Este algarismo de controlo é escolhido por forma a que a soma dos 9 algarismos seja um múltiplo de 10 e assim detetar erros de digitação. O campo com a data de nascimento tem o formato [dd-mm-aaaa]. Por questões de eficiência, economia de espaço de memória e posterior actualização dos dados, a naturalidade (local de nascimento do leitor) é representada apenas por um código, o *id_freguesia*. Naturalmente que, apesar deste código se referir à freguesia de onde o leitor é natural, ele permite também identificar os respetivos concelho e distrito. Assim, o *id_freguesia* é sempre constituído por 6 caracteres, sendo que os 2 caracteres iniciais são dígitos e identificam o distrito e os 2 seguintes são também dígitos e, em conjunto com os anteriores identificam o concelho. De notar que os 2 últimos caracteres podem ser letras e não algarismos. A lista dos distritos, dos concelhos e das freguesias é disponibilizada nos ficheiros de texto "*distritos.txt*", "*concelhos.txt*" e "*freguesias.txt*", respetivamente (ver a estrutura destes ficheiros abaixo).

requisitantes.txt			
<id_requisitante>\t<Nome>\t	<data_nasc>\t	<id_freguesia>	
140130623	Cristiano Pinto	11-11-1963	101209
...
768391411	Dulcineia Bacelar	16-08-2000	180605
...
223280724	Hugo Matos	29-09-1976	010304

distritos.txt		concelhos.txt		freguesias.txt	
<id_distrito>\t<Distrito>		<id_concelho>\t<Concelho>		<id_freguesia>\t<Freguesia>	
01	Aveiro	0101	Agueda	010103	Aguada de Cima
...
...	...	1821	Tondela	182102	Campo de Besteiros
...
18	Viseu	1824	Vouzela	182319	Orgens

3. Funcionalidades

Resumindo, o programa a implementar deve permitir diversas funcionalidades, acessíveis a partir de menus bem estruturados, nomeadamente:

1. Carregar os dados disponibilizados nos ficheiros para a memória, utilizando estruturas de dados convenientes (ter em atenção a memória utilizada e a facilidade de utilização na posterior implementação das restantes funcionalidades). Devem ser validados todos os dados lidos do ficheiro "*requisitantes.txt*" - O campo *id_requisitante* tem de ter 9 algarismos e a soma dos 9 algarismos tem de ser um múltiplo de 10; O campo *data_nasc* deve ter o formato [dd-mm-aaaa]; O campo *id_freguesia* tem de ter 6 caracteres e, os valores que estão no ficheiro têm de corresponder a freguesias efetivamente existentes. Dados com formato errado devem ser descartados, após ter registado o erro no ficheiro "*logs.txt*". Quando ocorre um erro, toda a linha deve ser copiada para o ficheiro "*logs.txt*", seguida de uma mensagem sobre o erro. Quando ocorre um erro, se não existir o ficheiro "*logs.txt*", este deve ser criado. Se já existir, deve ser acrescentada a informação do erro no fim, preservando o conteúdo anterior do ficheiro. Mesmo entre sucessivas execuções do programa, o ficheiro "*logs.txt*" nunca deve ser apagado. Caso já tenham ocorrido requisições de livros, os dados destas requisições devem também ser carregados do respetivo ficheiro para a memória;
2. Gravar a informação relativa aos livros, aos requisitantes e às requisições para ficheiros convenientes, tanto quando o utilizador o solicite como imediatamente antes do programa terminar;
3. Gerir os livros, permitindo:
 - a. Adicionar livros à Biblioteca (na área respetiva);
 - b. Listar todos os livros da Biblioteca (organizados por área);
 - c. Determinar qual a área com mais livros;
 - d. Verificar se um dado livro existe (dado o ISBN), se existir deve mostrar os seus dados;
 - e. Determinar qual(ais) os livros mais recentes (O que tiver o ano mais recente!);
 - f. Determinar qual o livro mais requisitado (Se existir mais do que um,

- deve devolver o primeiro encontrado);
 - g. Determinar qual área mais requisitada (Se existir mais do que uma, deve devolver a primeira encontrada);
4. Gerir a lista de requisitantes, permitindo:
- a. Inclusão de novos requisitantes;
 - b. Verificar se um requisitante existe na lista, dando o seu nome;
 - c. Listar todos os clientes, com uma ordenação à escolha do utilizador - por ordem alfabética do nome, por ordem do campo id_freguesia, por ordem alfabética do apelido (considera-se apelido a última palavra do nome completo);
 - d. Determinar a idade máxima de todos os requisitantes;
 - e. Determinar a média das idades de todos os requisitantes;
 - f. Contar o número de pessoas cuja idade é superior a um dado valor (introduzido pelo teclado);
 - g. Determinar qual a idade com mais requisitantes;
 - h. Mostrar as requisições de um dado requisitante;
 - i. Devolver uma lista das pessoas que nunca requisitaram nada;
 - j. Listar os requisitantes que no momento têm livros requisitados;
 - k. Determinar qual o sobrenome mais usado pelos requisitantes;
 - l. Determinar o número de pessoas de um dado Distrito (ou Concelho), cujo nome contém um determinado apelido/nome, por ex. determinar o número de pessoas do concelho 'Viseu' cujo nome contém 'Lopes';
- EXTRAS:
- m. Listar requisitantes que nasceram num Domingo (ou cujo aniversário num determinado ano é ao Domingo);
 - n. Listar requisitantes cujo aniversário, num determinado ano, é na quaresma (entre o Carnaval e a Páscoa), (ou que nasceram na quaresma).
5. Gerir as requisições de livros, permitindo:
- a. Efetuar uma requisição de um livro por parte de um requisitante;
 - b. Devolver um livro que anteriormente estava requisitado;
 - c. Listar os livros atualmente requisitados (organizados por área);
6. Determinar a memória ocupada por toda a Estrutura de Dados;
7. Gravar toda a informação para um dado ficheiro XML, cujo nome é introduzido pelo utilizador;
8. Libertar convenientemente toda a memória usada pelas estruturas de dados antes do programa terminar.

Nas diversas listagens a implementar, dada a potencial elevada quantidade de informação, deve implementar um mecanismo de avanço página a página e deve ser oferecida a possibilidade de gravar a listagem em ficheiro de texto *.txt* e *.csv*,

para posterior consulta num simples editor de texto (*Notepad*, *Notepad++*, *Sublime Text*, *TextMate* ou outro) e no excel;

As funcionalidades solicitadas devem ser implementadas por funções distribuídas de acordo com a funcionalidade, por exemplo:

- O código das funções necessárias para fazer o tratamento/*parse* dos ficheiros de texto devem estar nos ficheiros *texto.h* e *texto.c*.
- As funções necessárias para manipular listas ligadas devem estar nos ficheiros *lista.h* e *lista.c*.
- As funções necessárias para o *hashing* devem estar nos ficheiros *hashing.h* e *hashing.c*.
- ...

4. **Formato de Entrega**

O projeto deve ser desenvolvido em grupo. Cada grupo pode ter, no máximo, 4 alunos.

A entrega do projeto deve consistir num único ficheiro comprimido com o formato *ZIP*, enviado por um dos alunos do grupo para a página da unidade curricular na plataforma de *elearning* (<https://moodle.estgv.ipv.pt/>). Este ficheiro deve ter o nome "*ProjED_N1_N2_N3_N4.zip*", em que *N1*, *N2*, *N3* e *N4* são os números mecanográficos dos alunos do respetivo grupo (por exemplo, *ProjED_12033_16088_15765_15231.zip*) e deve incluir:

- Um ficheiro de texto com a identificação dos alunos do grupo (nome completo e número);
- Uma pasta com os ficheiros do código desenvolvido (ficheiros *.c* ou *.cpp* e *.h*);
- Uma pasta com um relatório que descreva o trabalho produzido, as estruturas de dados usadas e as funcionalidades implementadas.

O ficheiro *.zip* enviado não deverá conter quaisquer ficheiros executáveis.

5. **Data de Entrega**

- 02/06/2024; para avaliação em época normal (Avaliação ocorre no período das aulas de compensação)

- 02/07/2024; para avaliação em época recurso;

Caso se revele necessário, poderão ser feitas atualizações ou alterações a este enunciado, pelo que os alunos deverão estar atentos a esta eventualidade. Qualquer atualização ou alteração será devidamente anunciada na plataforma de *elearning* desta unidade curricular.