



Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Unidade Curricular: Algoritmos e Programação

Relatório Relativo ao Trabalho Prático

Tema: Biblioteca Interativa

Realizado por: Cassamo Abdul Latif – pv21373

Viseu, 2023

Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Relatório relativo ao Trabalho Prático

Curso de Licenciatura em Engenharia Informática

Unidade Curricular de Algoritmos e Programação

Biblioteca Interativa

Ano Letivo 2022/23

Viseu, 2023

RESUMO

Pretende-se implementar uma aplicação informática para uma Biblioteca Interativa, que permita fazer a gestão das publicações requisitadas pelos seus utentes.

A biblioteca funciona de seguinte forma: Na requisição de uma publicação, é guardada a informação da data de requisição e da sua validade (requisição não pode ultrapassar 30 dias). Assim que a publicação é devolvida, volta a ficar disponível. Caso não seja entregue na validade, é gerada uma multa por dia de incumprimento e o utente fica em incumprimento: primeiros 5 dias valor de 0.5 €/dia, 6º ao 10º dia valor de 1€/dia, a partir do 11º dia valor de 2€/dia, mas nunca ultrapassando os 50€. Se, no mês anterior, o utente tiver estado em incumprimento (pagou multa), a requisição só pode ser feita por 30 dias – nº dias em incumprimento no mês anterior. Em cada instante, pretende-se ter a informação relativa às Publicações. Cada Publicação é caracterizada pelo ISBN (Identificação única da Publicação), Título, Autor, Categoria (romance, Histórico, fantasia, etc.), Ano e estado da publicação (Disponível, Requisitado ou Indisponível). O utente é caracterizado pelo nome, nif, concelho e contacto.

Palavras-Chave: Biblioteca Interativa, requisição, publicação, multa, utente.

ÍNDICE

<i>1. Introdução</i>	<i>1</i>
<i>2. Tipos e Estruturas de dados</i>	<i>3</i>
2.1. Tipo de dados	3
2.2. Estrutura de dados	4
<i>3. Implementação das funcionalidades</i>	<i>5</i>
3.1. Letura de ficheiros e carregamento de dados para estruturas de dados convenientes	5
3.2. Inserção, alteração ou remoção dados de utentes, de publicações e de requisições;	5
3.3. Visualização da informação, em ficheiros de texto e/ou no ecrã, relativa a uma dada Publicação, autor, título, categoria de publicação, etc. (usar diferentes filtros);	6
3.4. Listar todos os utentes que já requisitaram um dado livro, por ordem alfabética do nome	7
3.5. Outras funcionalidades	7
<i>4. Organização dos arquivos de publicações e requisições</i>	<i>8</i>
<i>5. Conclusão</i>	<i>9</i>
<i>6. Referências</i>	<i>10</i>
<i>7. Bibliografia</i>	<i>11</i>

1. Introdução

Este trabalho consiste na implementação de uma aplicação informática para uma Biblioteca Interativa, que permita fazer a gestão das publicações requisitadas pelos seus utentes. O sistema servirá como auxílio na gestão das publicações requisitadas na biblioteca permitindo assim melhor controle das publicações.

O trabalho tem como objetivo geral, implementar uma aplicação informática para uma Biblioteca Interativa. E tem como objetivos específicos o cumprimento dos seguintes requisitos do sistema:

- a) Permitir leitura de ficheiros de dados, carregando-os para estruturas de dados convenientes (verificar existência de possíveis erros);
- b) Permitir inserção, alteração ou remoção dados de utentes, de publicações e de requisições;
- c) Permitir a visualização da informação, em ficheiros de texto e/ou no ecrã, relativa a uma dada Publicação, autor, título, categoria de publicação, etc. (usar diferentes filtros);
- d) Fornecer a informação das Publicações requisitadas por um dado Utente e o respetivo Estado;
- e) Dar o número total de incumprimentos e listá-los com nome e nif do cliente para cada ISBN do livro em incumprimento;
- f) Listar todos os utentes que já requisitaram um dado livro, por ordem alfabética do nome;
- g) Calcular o tempo médio de requisição, para uma dada categoria de livros;
- h) Calcular o(s) livro(s) mais requisitado(s) e o(s) utente(s) mais ativo(s).
- i) Listar os cancelhos, por ordem decrescente de número de utentes com requisições ativas.
- j) Listar os utentes, por ordem decrescente de valor a pagar de multa.
- k) Listar os próximos 5 livros a serem entregues, mas que não estejam em incumprimento

Este documento está organizado em cinco capítulos que se seguem a esta introdução.

No segundo capítulo aborda-se sobre os tipos e estruturas de dados necessários para a implementação do sistema.

Segue-se o capítulo três, onde aborda-se das principais funcionalidades do sistema e a função onde consta o código que a que implementa.

O capítulo quatro aborda XXX.

Termina-se com o capítulo cinco, onde se apresentam as conclusões deste trabalho.

2. Tipos e Estruturas de dados

Este capítulo aborda sobre os tipos de dados que foram criados para o sistema. Todo sistema informático manipula objetos (dados) para poder realizar suas tarefas.

2.1. Tipo de dados

Os tipos de dados por nós criados são os seguintes:

- **Publicação** – tipo de dados que armazena os dados de uma publicação que o utente pode se interessar e requisitar.
- **Requisição** – tipo de dado que armazena dados relacionados a uma requisição
- **Utente** – tipo de dado que armazena dados de utentes
- **Data** – Tipo de dado que armazena uma data. Usamos para armazenar as datas de uma requisição.

O corpo das estruturas em C é seguinte:

```
typedef struct Pubs {  
    char ISBN[24];  
    char Titulo[80];  
    char Autor[256];  
    char Categoria[32];  
    int Ano;  
    int estado;// 0 - disponivel, 1 - indisponivel  
} PUBLICACAO;
```

```
typedef struct requisicao {  
    char ISBN[24];  
    int nif;  
    DATA data_requisicao;  
    DATA data_devolucao;  
    int estado;  
} REQUISICAO;
```

```
typedef struct utente {  
    char Nome[80];  
    int nif;  
    char concelho[80];  
    int contacto;  
} UTENTE;
```

```
typedef struct {  
    int dia;  
    int mes;
```

```
    int ano;  
} DATA;
```

2.2. Estrutura de dados

Para armazenar um conjunto dos tipos de dados vistos na secção anterior usamos o **array** ou **vetor** como estrutura de dados. Em nossa implementação tem um vetor de Publicações, um de Requisições e um de Utentes, todos como variáveis globais.

3. Implementação das funcionalidades

Este capítulo aborda sobre as funcionalidades implementadas e as funções que contêm o código que implementa tais funcionalidades. Aqui considero como funcionalidades os requisitos mencionados na introdução.

3.1. *Letura de ficheiros e carregamento de dados para estruturas de dados convenientes*

Esta funcionalidade consiste na leitura de dados em arquivos e para depois armazená-los em estruturas de dados. Nós criamos uma função para carregar dados de cada tipo.

O protótipo das funções que realizam tal operação são:

```
void carregar_utente();  
void carregar_publicacao();  
void carregar_requisicao();
```

3.2. Inserção, alteração ou remoção dados de utentes, de publicações e de requisições;

Estas são as funcionalidades bases do sistema, pois permitem que ocorra o fluxo de dados. As funções de inserção permitem a inserção de um dado no sistema, a de alteração permite modificar um dado e a de remoção permite remover.

Os protótipos das funções para realizar tais operações em cada tipo de dado são:

- Para utente

```
void inserir_utente(UTENTE utente);  
void inserir_utente_front();  
int alterar_utente(int nif, UTENTE utente);  
int remover_utente(int nif);
```

- Para Publicação

```
void inserir_publicacao(PUBLICACAO publicacao)  
void alterar_publicacao(char * ISBN, PUBLICACAO publicacao)  
void remover_publicacao(char * ISBN)
```

-
- Para requisição

```
        inserir_requisicao(REQUISICAO requisicao)
void alterar_requisicao(char * ISBN, int nif, REQUISICAO requisicao)
void remover_requisicao(char * ISBN, int nif)
```

3.3. Visualização da informação, em ficheiros de texto e/ou no ecrã, relativa a uma dada Publicação, autor, título, categoria de publicação, etc. (usar diferentes filtros);

Esta funcionalidade permite a apresentação da informação no terminal. As funções que realizam tal operação para cada tipo de dado são:

- Para Utente:

```
void imprimir_lista_utentes(LISTA_UTENTE lista, int total);
void imprimir_todos_utentes()
```

- Para Publicação

```
void imprimir_lista_publicacoes(LISTA_PUBLICACOES lista, int total);
void imprimir_todas_publicacoes()
```

- Para Requisição

```
void imprimir_lista_requisicoes(LISTA_REQUISICAO lista, int total);
void imprimir_todas_requisicoes()
```

Para filtração da informação são usadas funções de filtragens como as seguintes:

- Para Autente

```
UTENTE pesquisar_utente_por_nif(int nif);

LISTA_UTENTE pesquisar_utente_por_nome(char * nome, int * nr_elementos)
```

Funções com protótipos semelhantes têm disponíveis para os outros tipos de dados também (Publicação e Requisição).

3.4. Listar todos os utentes que já requisitaram um dado livro, por ordem alfabética do nome

Essa lista é obtida através da função com seguinte protótipo:

```
void listar_utentes_por_livro(char *ISBN_desejado);
```

3.5. Outras funcionalidades

Outras funcionalidades podem ser tidas a partir da combinação das funções com os seguintes protótipos:

```
DATA somar_n_dias(int n);  
float calcular_multa(REQUISICAO req);  
int pub_disponivel(char * ISBN);  
void print(char **opcoes, int tamanho);  
void adicionar_publicacao_lista(LISTA_PUBLICACAO lista_resultado,  
    PUBLICACAO publicacao, int *tamanho_lista_resultado);  
void carregar_requisicao();  
void gravar_dados_requisicao();  
DATA data_hoje();  
void gravar_dados_requisicao();  
void gravar_dados_utentes();  
void gravar_dados_publicacoes();
```

4. Organização dos arquivos de publicações e requisições

Fiz uma alteração no arquivo de publicações onde acrescentei a coluna estado. No final o arquivo ficou da seguinte maneira:

ISBN	Titulo	Autor	Categoria	Ano	estado
4628-4149-890-2159	Mortal Inquisition	Erick Barros	Drama	1925	1
4709-8805-4688-1497	Solar Attack	Alexandre Araujo	Drama	1943	0

A estrutura do arquivo de requisições é a seguinte:

ISBN	nif	dia_reqmes_req	ano_req	dia_dev	mes_dev	ano_dev
7680-4859-3240-334	248998788	1	12	2021	20	12
						2021
						1

5. Conclusão

A maioria dos requisitos do sistema estão implementados e perfeitamente funcionais.

6. Referências

7. Bibliografia

