

Universidade da Beira Interior

Departamento de Informática



Departamento de
Informática

Software de Gestão de uma Livraria

Elaborado por:

Nuno Marques nº38958
Tomás Vicente nº39067

Professores:

Professor Doutor Carlos Barrico
Professor Doutor Frutuoso Silva
Professor Doutor José Morgado

30 de maio de 2021

Conteúdo

Conteúdo	0
Lista de Figuras	3
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	1
1.3 Organização do Documento	1
2 Estruturas de Dados	3
2.1 Introdução	3
2.2 Livro	3
2.3 Cliente	4
2.4 Encomenda	4
3 Menus Principais	5
3.1 Introdução	5
3.2 Menus Principais	5
3.3 Menus Secundários	8
4 Funções Auxiliares	9
4.1 Introdução	9
4.2 Funções Auxiliares <i>Livros.h</i>	9
4.3 Funções Auxiliares <i>Cliente.h</i>	12
4.4 Funções Auxiliares <i>Encomenda.h</i>	14
5 Funcionalidades Extra	15
5.1 Introdução	15
5.2 Funcionalidades Extra	15
5.2.1 Consultar Livros	15
5.2.2 Consultar Clientes	15
5.2.3 Consultar Encomendas	16
5.2.4 Menu de Operações	16

CONTEÚDO	1
<hr/>	
6 Conclusões	17
6.1 Conclusões	17

Lista de Figuras

3.1	Menu Principal	5
3.2	Menu de Ficheiros	6
3.3	Menu de Livros	6
3.4	Menu de Clientes	6
3.5	Menu de Encomendas	7
3.6	Menu de Operações	7
3.7	Menu Secundário - Consulta de Livros	8
3.8	Menu Secundário - Consulta de Clientes	8

Lista de Excertos de Código

2.1	Estrutura Livro.	3
2.2	Estrutura Cliente.	4
2.3	Estrutura Encomenda.	4

Capítulo

1

Introdução

1.1 Enquadramento

Este relatório foi feito no contexto da unidade curricular de Algoritmos e Estruturas de Dados do segundo ano da licenciatura em Engenharia Informática da Universidade da Beira Interior.

1.2 Objetivos

O principal objetivo deste projeto, foi a implementação de um software de gestão para uma livraria, efetuado com a linguagem de programação C.

1.3 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. Primeiro capítulo – **Introdução** – É apresentado o enquadramento, os objetivos do projeto e a organização do relatório, para uma fácil navegação.
2. Segundo capítulo – **Estruturas de Dados** – São apresentadas as estruturas de dados utilizadas para este projeto.
3. Terceiro capítulo – **Menus Principais** – São apresentados os menus principais da aplicação.

4. Quarto capítulo – **Funções Auxiliares** – São apresentadas todas funções auxiliares deste projeto.
5. Quinto Capítulo – **Funcionalidades Extra** – São apresentadas as funcionalidades extra deste projeto.
6. Sexto capítulo – **Conclusões** – São apresentadas as conclusões deste projeto.

Capítulo

2

Estruturas de Dados

2.1 Introdução

Neste capítulo são apresentadas as três estruturas de dados utilizadas neste projeto.

2.2 Livro

O trecho de código seguinte mostra a estrutura de dados Livro:

```
typedef struct Livro {  
    int ISBN;  
    char Titulo [50];  
    char Idioma [100];  
    char PrimeiroAutor [100];  
    char SegundoAutor [100];  
    char Editora [100];  
    int Ano;  
    char AreaCientifica [50];  
    float Preco;  
    int Quantidade;  
} LIVRO;
```

Excerto de Código 2.1: Estrutura Livro.

2.3 Cliente

O trecho de código seguinte mostra a estrutura de dados Cliente:

```
typedef struct Cliente {  
    int NIF;  
    char Nome[100];  
    char Morada[100];  
    int Telefone;  
    PNodeFila ListaCompras;  
}CLIENTE;
```

Excerto de Código 2.2: Estrutura Cliente.

2.4 Encomenda

O trecho de código seguinte mostra a estrutura de dados Encomenda:

```
typedef struct Encomenda {  
    int id;  
    int ClienteNIF;  
    int LivroISBN;  
    char DataEncomenda[10];  
    int UnidadesEncomendadas;  
    float PrecoTotal;  
}ENCOMENDA;
```

Excerto de Código 2.3: Estrutura Encomenda.

Capítulo

3

Menus Principais

3.1 Introdução

Neste terceiro capítulo são apresentados os menus principais e secundários da aplicação, responsáveis pela boa navegação do utilizador pela aplicação.

3.2 Menus Principais

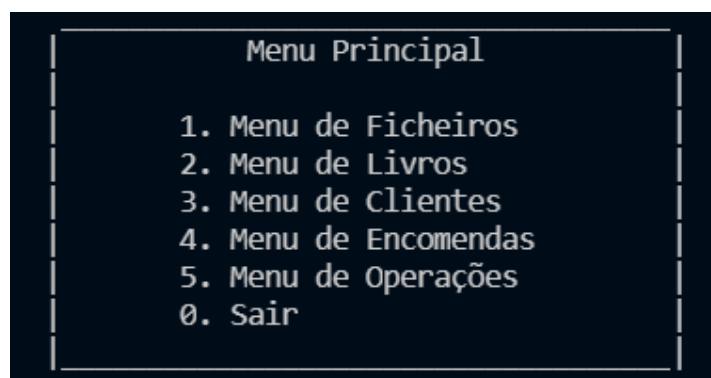


Figura 3.1: Menu Principal

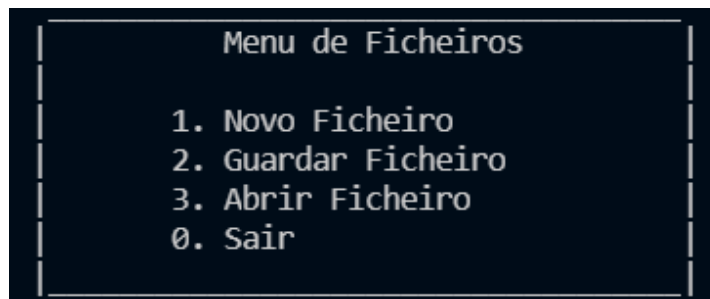


Figura 3.2: Menu de Ficheiros

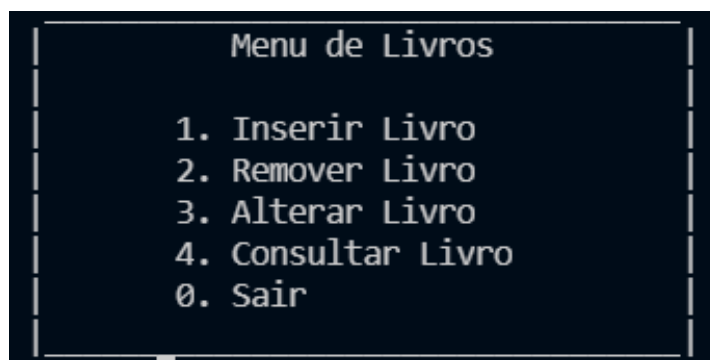


Figura 3.3: Menu de Livros

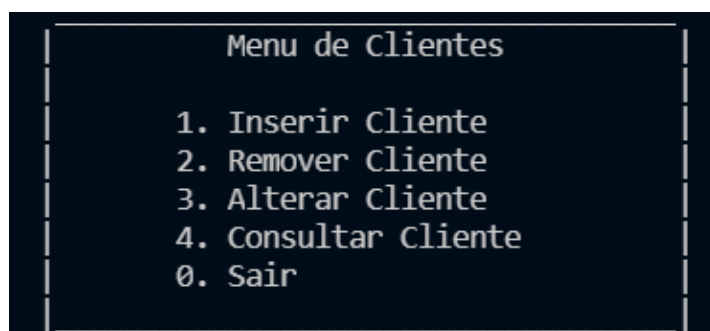


Figura 3.4: Menu de Clientes

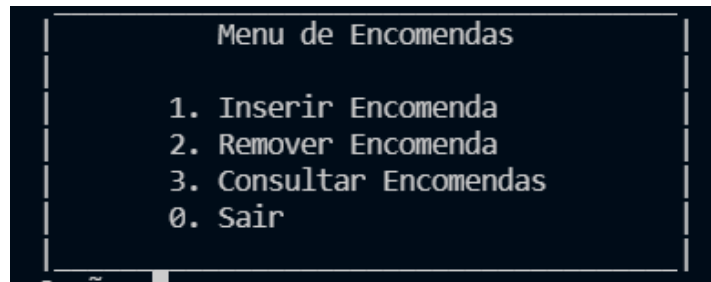


Figura 3.5: Menu de Encomendas

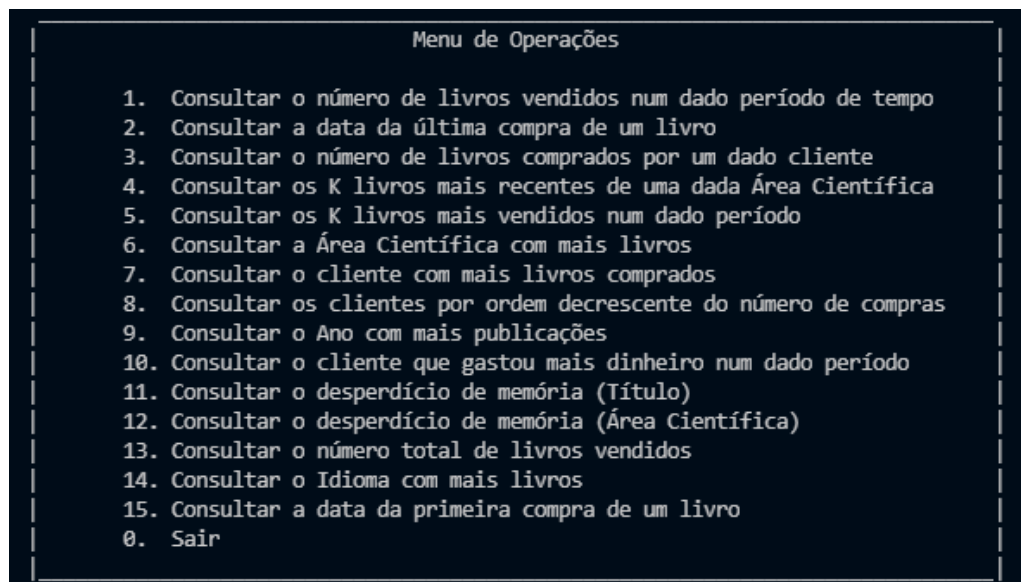


Figura 3.6: Menu de Operações

3.3 Menus Secundários

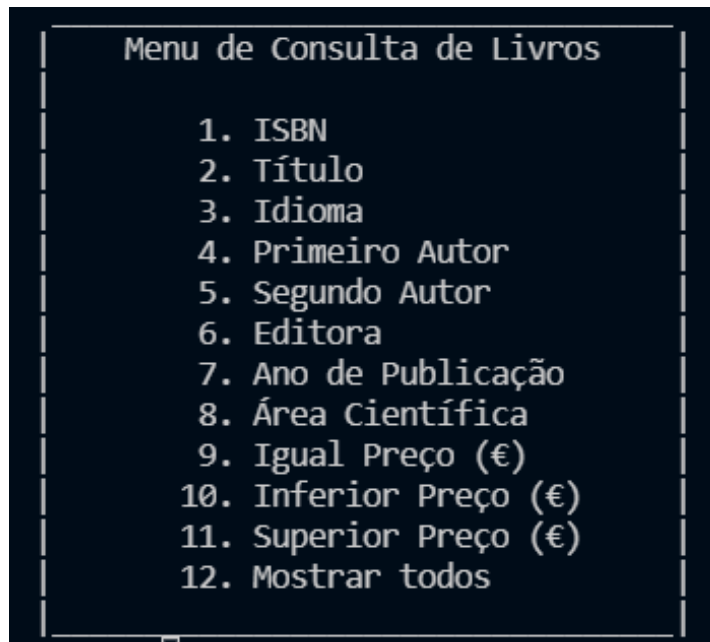


Figura 3.7: Menu Secundário - Consulta de Livros

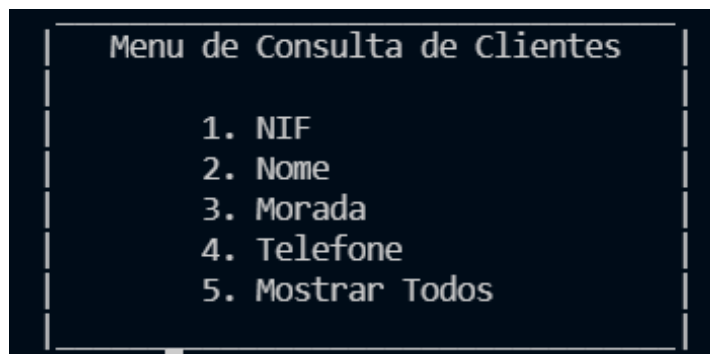


Figura 3.8: Menu Secundário - Consulta de Clientes

Capítulo

4

Funções Auxiliares

4.1 Introdução

Neste quarto capítulo do relatório, são apresentadas todas as funções auxiliares criadas no projeto, para uma melhor leitura e organização do sistema. Estas funções dividem-se por três ficheiros distintos: *Livros.h*, *Cliente.h* e *Encomenda.h*. À frente do cabeçalho de cada função é apresentado um breve resumo sobre para que serve a função.

4.2 Funções Auxiliares *Livros.h*

1. **PNodo CriarNodo (LIVRO X)** – Utilizada para criar um nodo do tipo LIVRO;
2. **void InserirPrimeiroElemento (PNodo *Head, PNodo *Tail, LIVRO X)** – Utilizada para inserir o primeiro elemento na lista, a *Head* fica igual à *Tail* porque apenas existe um elemento na lista.
3. **PNodo InserirCabeca (PNodo Head, LIVRO X)** – Utilizada para inserir elementos na cabeça da lista.
4. **void ConsultarLivro(LIVRO X)** – Utilizada para exibir no teclado toda a informação referente a um livro.
5. **void Listar (PNodo Head)** – Utilizada para listar todos os livros existentes na lista.
6. **int Tamanho (PNodo Head)** – Utilizada para verificar quantos livros existem na lista.

7. **PNodo LibertarNodo (PNodo P)** – Utilizada para libertar um nodo.
8. **int CompararElementosISBN (LIVRO X, LIVRO Y)** – Utilizada para comparar dois elementos do tipo livro, por ISBN.
9. **PNodo Procurar (PNodo Head, LIVRO X)** – Utilizada para verificar se um dado livro X, existe na lista de livros.
10. **int TemLivro(PNodo Head, LIVRO L)** – Utilizada também para verificar se um dado livro L existe na lista de livros, mas retorna apenas se existe ou não.
11. **LIVRO ProcurarLivro(PNodo Head, LIVRO L)** – Utilizada para procurar um livro na lista de livros.
12. **void RemoverUnicoElemento (PNodo *Head, PNodo *Tail)** – Utilizada para remover o único elemento existente na lista, no caso da lista só possuir um elemento.
13. **PNodo RemoverCabeca (PNodo Head)** – Utilizada para remover o elemento que se encontra na cabeça da lista.
14. **PNodo RemoverCauda (PNodo Tail)** – Utilizada para remover o elemento que se encontra na cauda da lista.
15. **PNodo RemoverMeio (PNodo Head, LIVRO X)** – Utilizada para remover um elemento que se encontre no meio da lista, ou seja, não está nem na cabeça, nem da cauda.
16. **int ConsultarCabeca (PNodo Head, LIVRO X)** – Utilizada para verificar se o elemento X, se encontra na cabeça da lista (função auxiliar, utilizada na próxima função).
17. **void RemoverLivro(PNodo *Head, PNodo *Tail, int isbn)** – Utilizada para remover um livro, dado o seu ISBN, sendo este único para cada livro.
18. **void ListarISBN (PNodo Head, int isbn)** – Utilizada para listar o livro que possui o ISBN dado.
19. **int CompararElementosAno (LIVRO X, LIVRO Y)** – Utilizada para comparar o ano de publicação entre dois livros. Se o ano do livro X for superior ao do livro Y retorna 1, caso contrário retorna -1.
20. **void ListarAno (PNodo Head, int ano)** – Utilizada para listar os livros dado um determinado ano.

21. **int CompararElementosPreco (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pelo preço. Se o preço do livro X for superior ao do livro Y, retorna 1, caso contrário retorna -1.
22. **void ListarPreco (PNodo Head, int preco, int flag)** – Utilizada para listar os elementos que possuam um preço igual, inferior ou superior ao fornecido, dependendo da variável flag.
23. **int CompararElementosTitulo (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pelo título.
24. **void ListarTitulo (PNodo Head, char *tmp)** – Utilizada para listar os livros cujo título, ou parte dele, corresponde ao título fornecido pelo utilizador.
25. **int CompararElementosIdioma (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pelo seu idioma.
26. **void ListarIdioma (PNodo Head, char *tmp)** – Utilizada para listar os livros, cujo idioma seja igual ao idioma fornecido pelo utilizador.
27. **int CompararElementosPAutor (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pelo seu autor principal.
28. **void ListarPAutor (PNodo Head, char *tmp)** – Utilizada para listar os livros, cujo autor principal seja igual ao autor principal fornecido pelo utilizador.
29. **int CompararElementosSAutor (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pelo seu autor secundário.
30. **void ListarSAutor (PNodo Head, char *tmp)** – Utilizada para listar os livros, cujo autor secundário seja igual ao autor secundário fornecido pelo utilizador.
31. **int CompararElementosEditora (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pela sua editora.
32. **void ListarEditora (PNodo Head, char *tmp)** – Utilizada para listar os livros, cuja editora seja igual à editora fornecida pelo utilizador.
33. **int CompararElementosAreaCientifica (LIVRO X, LIVRO Y)** – Utilizada para comparar dois livros, pela sua área científica.
34. **void ListarAreaCientifica (PNodo Head, char *tmp)** – Utilizada para listar os livros, cuja área científica seja igual à fornecida pelo utilizador.

35. **void ListarACeANO (PNode Head, char *tmp, int ano)** – Utilizada para listar os livros de uma determinada área científica e de um determinado ano específico.

4.3 Funções Auxiliares *Cliente.h*

1. **PNodeAB CriarAB()** – Utilizada para criar uma Árvore Binária.
2. **PNodeAB CriarNodoAB(CLIENTE X)** – Utilizada para criar um nodo cliente, na árvore binária.
3. **int NumeroNodosAB (PNodeAB T)** – Utilizada para verificar o número de nós da árvore binária.
4. **int CompararElementosNIF (CLIENTE X, CLIENTE Y)** – Utilizada para comparar dois clientes, pelo NIF.
5. **PNodeAB InserirABP (PNodeAB T, CLIENTE X)** – Utilizada para inserir um cliente, na árvore binária de pesquisa.
6. **CriarABPEquilibradaIB (PNodeAB T)** – Utilizada para criar uma árvore binária de pesquisa equilibrada.
7. **void ABPEqInsercaoBinaria (PNodeAB *TE, CLIENTE L[], int inicio, int fim)** – Utilizada na função CriarABPEquilibradaIB.
8. **void CriarSequenciaEmOrdem (PNodeAB T, CLIENTE L[], int *N)** – Utilizada na função CriarABPEquilibradaIB.
9. **int PesquisarABP (PNodeAB T, CLIENTE X)** – Utilizada para pesquisar um dado cliente na árvore.
10. **CLIENTE PesquisarClienteAlterar(PNodeAB T, CLIENTE C)** – Utilizada para pesquisar um dado cliente na árvore e se existe, retorna esse mesmo cliente.
11. **void MostrarCliente (CLIENTE X)** – Utilizada para mostrar todas as informações relativas a um cliente.
12. **void ListarEmOrdemAB (PNodeAB T)** – Utilizada para listar os clientes por ordem crescente de NIF.
13. **PNodeAB LibertarNodoAB(PNodeAB P)** – Utilizada para libertar um nó da árvore.

14. **PNodeAB RemoverABP (PNodeAB T, CLIENTE X)** – Utilizada para remover um cliente da árvore.
15. **PNodeAB SubstituirNodeDireita (PNodeAB T, CLIENTE *X)** – Utilizada na função RemoverABP.
16. **PNodeAB RemoverNodeABP (PNodeAB T)** – Utilizada na função RemoverABP.
17. **void ListarNIF (PNodeAB T, CLIENTE X)** – Utilizada para listar o cliente que possui o NIF igual ao fornecido.
18. **int CompararElementosNome(CLIENTE X, CLIENTE Y)** – Utilizada para comparar dois clientes, por nome.
19. **void ListarNome(PNodeAB T, CLIENTE C)** – Utilizada para listar os clientes cujo nome é igual ao nome fornecido.
20. **int CompararElementosMorada(CLIENTE X, CLIENTE Y)** – Utilizada para comparar dois clientes, pela sua morada.
21. **void ListarMorada(PNodeAB T, CLIENTE C)** – Utilizada para listar os clientes cuja morada é igual à fornecida.
22. **int CompararElementosTelefone (CLIENTE X, CLIENTE Y)** – Utilizada para comparar dois clientes, pelo seu telefone.
23. **void ListarTelefone(PNodeAB T, CLIENTE C)** – Utilizada para listar os clientes cujo número de telefone é igual ao fornecido.
24. **void MostrarNIFNOME(CLIENTE X)** – Utilizada para mostrar o Nome e o NIF de um cliente.
25. **void ListarAllNIFNOME(PNodeAB T)** – Utilizada para mostrar o Nome e o NIF de todos os clientes, ordenados por crescente do NIF.
26. **void ListarListaCompras(PNodeAB T, CLIENTE X)** – Utilizada para listar a lista de compras de um dado cliente, dado um NIF.
27. **void ListarNIFNOMEMaisLivrosComprados(PNodeAB T, CLIENTE X)** – Utilizada para listar o cliente com mais livros comprados.
28. **void SaveFile(PNodeAB T, FILE *fpC)** – Utilizada para guardar a estrutura de Clientes num ficheiro.

4.4 Funções Auxiliares *Encomenda.h*

1. **PNodeFila CriarNodeFila(ENCOMENDA X)** – Utilizada para criar um nodo na fila de encomendas.
2. **int FilaVazia(PNodeFila Fila)** – Utilizada para verificar se a fila se encontra vazia.
3. **PNodeFila Juntar(ENCOMENDA X, PNodeFila Fila)** – Utilizada para juntar uma encomenda no fim da fila.
4. **void ListarEncomenda(ENCOMENDA X)** – Utilizada para listar todas as informações referentes a uma encomenda.
5. **void MostrarFila(PNodeFila Q)** – Utilizada para listar todas as encomendas presentes na fila de encomendas.
6. **PNodeFila LibertarNodeFila(PNodeFila P)** – Utilizada para libertar um nodo da fila de encomendas.
7. **PNodeFila RemoverEncomenda(PNodeFila Fila, int id)** – Utilizada para remover uma encomenda da fila.
8. **int UltimoIDEncomenda(PNodeFila Q)** – Utilizada para retornar o id da última encomenda efetuada.
9. **int TemEncomenda(PNodeFila Q, int id)** – Utilizada para verificar se um determinado id de encomenda, existe na fila de encomendas.
10. **int LivrosVendidos(PNodeFila Q)** – Utilizada para verificar o numero total de livros vendidos.

Capítulo

5

Funcionalidades Extra

5.1 Introdução

Neste capítulo são apresentadas as funcionalidades extra da aplicação. Estas funcionalidades não eram requeridas para este projeto, foram implementadas por se achar que contribuíam para um projeto mais coeso e com mais informação disponível.

5.2 Funcionalidades Extra

5.2.1 Consultar Livros

No menu de consulta de livros foram adicionadas três opções extra, que considerámos importantes e que acrescentavam uma informação relevante para o dono da livraria.

1. **Consultar os Livros com o preço inferior a X** – Esta opção de consulta, lista todos os livros, cujo preço seja inferior a X.
2. **Consultar os Livros com o preço superior a X** – Esta opção de consulta, lista todos os livros, cujo preço seja superior a X.
3. **Mostrar Todos** – Esta opção de consulta, lista todos os livros existentes na livraria.

5.2.2 Consultar Clientes

No menu de consulta de clientes foi adicionada uma funcionalidade extra, que achamos ser crucial para um software de gestão de uma livraria.

1. **Mostrar Todos** – Esta opção de consulta, lista todos os clientes registrados na livraria.

5.2.3 Consultar Encomendas

No menu de consulta de encomendas foi também adicionada uma funcionalidade extra, igualmente crucial para um software de gestão de uma livraria.

1. **Consultar Encomendas** – Esta opção de consulta, lista todas as encomendas que foram efetuadas até ao momento na livraria.

5.2.4 Menu de Operações

No menu de operações foram implementadas quatro funcionalidades extra, que estão abaixo documentadas.

1. **Consultar o Desperdício de memória (Área Científica)**
2. **Consultar o número total de livros vendidos**
3. **Consultar o Idioma com mais livros**
4. **Consultar a data da primeira compra de um livro**

Capítulo

6

Conclusões

6.1 Conclusões

No decorrer deste projeto, colocámos à prova, muitas das componentes aprendidas nesta unidade curricular, como estruturas, filas, árvores binárias, etc. Conseguímos implementar todas as funcionalidades básicas pedidas pelo professor e ainda acrescentámos funcionalidades extra, na aplicação da livraria.

Concluímos que foi um projeto que contribuiu para o nosso desenvolvimento como programadores da linguagem C e que pode ser futuramente implementado numa livraria.

