

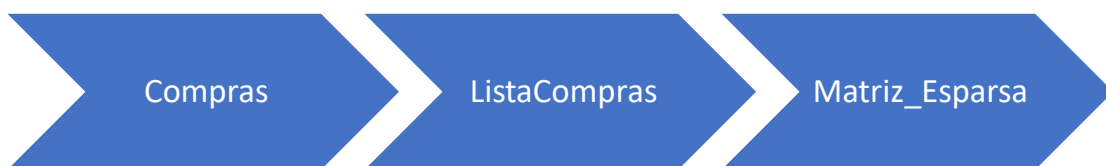
TRABALHO PRÁTICO I

LARISSA ISABELLE 3871
OTÁVIO SANTOS 3890
PEDRO CARDOSO 3877

RELATÓRIO

Para a implementação do trabalho proposto, decidimos começar a programação da parte interna para fora, de forma que primeiro foi primeiro programado o TAD Compras, logo em seguida o TAD ListaCompras que fez uso das operações implementadas pelo TAD Compras e por fim o TAD Matriz_Esparsa foi implementado fazendo uso das operações disponíveis pelo TAD ListaCompras. Essas especificações foram discutidas e colocadas no papel pelo grupo antes de iniciar a programação do programa.

Seguindo as especificações definidas pelo trabalho, a estrutura interna do trabalho ficou dessa forma:



Em Compras foram definidos os seguintes atributos:

```
1  typedef struct {
2      char data[10];
3      int quantidade;
4  } TCompras;
```

E as seguintes operações:

```
6  void NovaCompra(TCompras* pTCompras, char data[10], int quantidade);
7
8  void Imprime(TCompras* pTCompras);
```

Em ListaCompras os atributos foram os seguintes:

```
3  typedef struct CelulaC {
4      TCompras compra;
5      struct CelulaC* proximo;
6  } CelulaCompras;
7
8  typedef struct {
9      CelulaCompras* primeiro;
10     CelulaCompras* ultimo;
11     int Lquantidade;
12 } LCompras;
```

E as operações:

```
14 void FLVazia(LCompras* pLCompras);
15
16 void LInsere(LCompras* pLCompras, TCompras* pCompra);
17
18 int LEhVazia(LCompras* pLCompras);
19
20 void LImprime(LCompras* pLCompras);
```

No TAD Matriz_Esparsa, esses foram os atributos:

```
3 typedef struct CelulaM {
4     struct CelulaM* direita;
5     struct CelulaM* abaixo;
6     int linha;
7     int coluna;
8     LCompras compras;
9 } CelulaMatriz;
10
11 typedef struct {
12     int qtd_linhas;
13     int qtd_colunas;
14
15     CelulaMatriz* primLinha;
16     CelulaMatriz* primColuna;
17 } MatrizEsparsa;
```

Quanto às operações, o TAD apresenta as seguintes:

```
19 void LeMatriz (MatrizEsparsa* pMatriz, FILE* arquivo);
20
21 void InsereMatriz(MatrizEsparsa* pMatriz, LCompras* pLCompras, int linha, int coluna);
22
23 void FechaMatriz(MatrizEsparsa* pMatriz);
24
25 int MEhVazia(MatrizEsparsa* pMatriz);
26
27 void ImprimeMatriz (MatrizEsparsa* pMatriz);
28
29 int QuantidadeComprasPorProduto(MatrizEsparsa* pMatriz, int produto);
30
31 int QuantidadeComprasPorCliente(MatrizEsparsa* pMatriz, int cliente);
```

Dessa forma, todos os TAD's foram encapsulados e apenas são utilizados por meio de suas respectivas operações.

Ressaltamos que no TAD Matriz_Esparsa, algumas operações extras foram feitas para que pudessem ser utilizados pelo próprio TAD. Dessa forma, o

usuário não possui acesso às operações InsereMatriz, FechaMatriz e MehVazia, uma vez que elas apenas servem de controle interno do funcionamento do Programa.

Durante a programação foram utilizados alguns arquivos .txt para testes do funcionamento. Sendo esse o mais utilizado:

```
3 3
1 1 17/06/2000 4 21/07/2005 7 13/09/2014 1
1 3 15/08/2001 1 23/08/2007 4 21/12/2012 0 04/06/2008 7
2 1 14/09/2006 3
2 2 13/06/2011 8 25/12/2012 3
3 2 25/12/2012 2
3 3 25/12/2012 3 15/08/2001 1 23/08/2007 4 21/12/2012 0 04/06/2008 7
```

É importante dizer que o programa funciona somente se o arquivo de entrada estiver formatado exatamente dessa forma, que foi a especificada pela documentação do trabalho.

Os arquivos .txt utilizados como testes estão na pasta junto com esse documento, com os respectivos nomes: TesteUM.txt, TesteDOIS.txt e TesteTRES.txt.

Assim, foi possível perceber que uma boa conversa e definição de quais Tipos Abstratos de Dados e quais operações seriam criadas antes mesmo de começar a programar foi de suma importância para que o trabalho pudesse ser desenvolvido de forma organizada e com menos contratempos.