

C:\Users\avlal\OneDrive - Instituto Politécnico do Porto\2 - ESTG - Escola Superior Tecnologia Gestão\1º Ano\1º Semestre\4 - Introdução à Programação\5 - Projeto em grupo - Epoca Exames\LojaRC\src\lojarc\LojaRC.java

```
1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5 package lojarc;
6
7 import java.io.BufferedReader;
8 import java.io.File;
9 import java.io.FileNotFoundException;
10 import java.io.FileReader;
11 import java.io.IOException;
12 import java.io.PrintWriter;
13 import java.util.Scanner;
14
15
16 /**
17  * INTRODUÇÃO
18  *
19  * Trabalho realizado segundo os temas apresentados na UC
20  * IDE Netbeans
21  *
22  * GRUPO: LSIG_EF_10
23  * @author AvelinoAlmeida
24  */
25
26 public class LojaRC {
27
28     /**
29      * CONSTANTES
30      * Tamanho de incremente de "espaço de array"
31      * Tamanho inicial de cada array
32      * Tamanho maximo possivel
33      *
34      */
35     static final int INCREMENTO = 2;
36     static final int TAMANHO_INICIAL_CLIENTES = 10;
37     static final int TAMANHO_MAXIMO_CLIENTES = 50;
38     static final int TAMANHO_INICIAL_PRODUTOS = 10;
39     static final int TAMANHO_MAXIMO_PRODUTOS = 50;
40     static final int TAMANHO_INICIAL_COMPONENTES = 10;
41     static final int TAMANHO_MAXIMO_COMPONENTES = 50;
42     static final int TAMANHO_INICIAL_ENCOMENDAS = 10;
43     static final int TAMANHO_MAXIMO_ENCOMENDAS = 50;
44
45     //-----
46     //----- Gestão de Clientes -----//
47 }
```

```

48  /**
49   * Objetivo ler os dados dos clientes a partir de um arquivo "clientes.txt"
50   * @param clientes
51   * @throws FileNotFoundException
52   */
53 static void LerFicheiroClientes(ListaClientes clientes) throws FileNotFoundException{
54     File f = new File("clientes.txt");
55     Scanner s = new Scanner(f);
56     Cliente temp = new Cliente();
57     while(s.hasNextLine()){
58       String linha = s.nextLine();
59       String[] valores = linha.split(";");
60       if(valores[0].equals("CLIENTE")){
61         temp = new Cliente();
62         temp.id = Integer.parseInt(valores[1]);
63         temp.NIF = Integer.parseInt(valores[2]);
64         temp.nome = valores[3];
65         temp.morada = valores[4];
66         temp.pais = valores[5];
67         temp.estado = valores[6];
68         adicionarCliente(clientes, temp);
69         nCliente = temp.id+1;
70       }
71     }
72 }
73 /**
74  * Forma encontrada para gear um codigo ou id unico
75 */
76 static int nCliente = 1;
77
78 /**
79  * O método "criarFicheiroClientes" tem por finalidade
80  * escrever os dados dos clientes em um arquivo de texto.
81  * @param clientes
82  * @throws IOException
83 */
84 static void criarFicheiroClientes(ListaClientes clientes) throws IOException{
85   File f = new File("clientes.txt");
86   BufferedWriter wr = new BufferedWriter(new FileWriter(f));
87   for(int i=0;i<clientes.n_clientes;i++){
88     wr.write("CLIENTE;" +
89       +clientes.l_clientes[i].id+";" +
90       +clientes.l_clientes[i].NIF+";" +
91       +clientes.l_clientes[i].nome+";" +
92       +clientes.l_clientes[i].morada+";" +
93       +clientes.l_clientes[i].pais+";" +
94       +clientes.l_clientes[i].estado+"\n");
95   }
96   wr.flush();
97   wr.close();
98 }

```

```

99
100 /**
101 * O método pesquisarCliente é uma função que procura por um cliente na
102 * lista de clientes (ListaClientes) com base em seu NIF
103 * @param clientes
104 * @param nif
105 * @return
106 */
107 static int pesquisarCliente(ListaClientes clientes, int nif) {
108     int pos = -1,i=0;
109     while(pos == -1 && i< clientes.n_clientes ){
110         if(clientes.l_clientes[i].NIF == nif){
111             pos = i;
112         }
113         i++;
114     }
115     return pos;
116 }
117
118 /**
119 * Responsável por adicionar um novo cliente à lista de clientes.
120 * @param clientes
121 * @param novo
122 * @return true ou false
123 */
124 static boolean adicionarCliente(ListaClientes clientes, Cliente novo) {
125     int pos = pesquisarCliente(clientes, novo.id);
126     if (pos != -1) {
127         return false;
128     }
129
130     if (clientes.n_clientes == clientes.l_clientes.length) {
131         if (clientes.n_clientes != TAMANHO_MAXIMO_CLIENTES) {
132             Cliente[] temp = new Cliente[clientes.n_clientes + INCREMENTO];
133             for (int i = 0; i < clientes.n_clientes; i++) {
134                 temp[i] = clientes.l_clientes[i];
135             }
136             clientes.l_clientes = temp;
137         } else {
138             return false;
139         }
140     }
141     clientes.l_clientes[clientes.n_clientes++] = novo;
142     return true;
143 }
144
145 /**
146 * Esta função é responsável por verificar se já existe algum cliente
147 * com o NIF fornecido
148 * @param clientes
149 * @param NIF

```

```

150     * @return
151     */
152     static boolean verificarNIF(ListaClientes clientes, int NIF){
153         for(int i = 0; i < clientes.n_clientes;i++){
154             if(clientes.l_clientes[i].NIF == NIF){
155                 return true;
156             }
157         }
158         return false;
159     }
160
161 /**
162 * Imprime informações sobre um cliente com base no seu NIF da lista clientes.
163 * É feito por meio de um loop que percorre todos os clientes da lista
164 * @param clientes
165 * @param nif
166 */
167 static void LocalizarCliente(ListaClientes clientes, int nif){
168     for(int i = 0 ; i < clientes.n_clientes;i++ ){
169         if(clientes.l_clientes[i].NIF == nif){
170             System.out.println("-----+");
171             System.out.println("    Cliente ID: "+ clientes.l_clientes[i].id);
172             System.out.println(" |-----|");
173             System.out.println("  NIF   | "+clientes.l_clientes[i].NIF);
174             System.out.println("  Nome  | "+clientes.l_clientes[i].nome);
175             System.out.println("  Morada | "+clientes.l_clientes[i].morada);
176             System.out.println("  País   | "+clientes.l_clientes[i].pais);
177             System.out.println("  Estado  | "+clientes.l_clientes[i].estado);
178             System.out.println("-----+");
179         }
180     }
181 }
182
183 /**
184 * Função que permite a edição de um cliente existente numa lista de
185 * clientes.
186 * A função tem por entrada uma instância de Scanner e uma instância da
187 * classe ListaClientes
188 * @param sc
189 * @param clientes
190 */
191 static void editarClientes(Scanner sc, ListaClientes clientes) {
192     System.out.print("Digite o NIF do cliente a ser editado: ");
193     int nif = sc.nextInt();
194     int pos = pesquisarCliente(clientes, nif);
195     if (pos == -1) {
196         System.out.println("Cliente não encontrado.");
197         return;
198     }
199     Cliente cliente = clientes.l_clientes[pos];
200     int editar;

```

```

201     do {
202         System.out.println("+-----+");
203         System.out.println(" |      Editar Cliente   | ");
204         System.out.println("+-----+");
205         System.out.println(" | 1 | Nome           | ");
206         System.out.println(" | 2 | Morada          | ");
207         System.out.println(" | 3 | País            | ");
208         System.out.println(" | 0 | MENU            | ");
209         System.out.println("+-----+");
210         System.out.print("Escolha uma das seguintes opções: ");
211         editar = sc.nextInt();
212         switch (editar) {
213             case 1:
214                 System.out.print("Insira o novo nome do cliente: ");
215                 sc.nextLine();
216                 cliente.nome = sc.nextLine();
217                 break;
218             case 2:
219                 System.out.print("Insira a nova morada do cliente: ");
220                 sc.nextLine();
221                 cliente.morada = sc.nextLine();
222                 break;
223             case 3:
224                 System.out.print("Insira o novo país do cliente: ");
225                 sc.nextLine();
226                 cliente.pais = sc.nextLine();
227                 break;
228             case 0:
229                 break;
230             default:
231                 System.out.println("Opção inválida.");
232         }
233     } while (editar != 0);
234 }
235
236 /**
237 * Função remove um cliente da lista de clientes em função do NIF.
238 * @param lc
239 * @param nif
240 * @return
241 */
242 static boolean removerCliente(ListaClientes lc, int nif){
243     int pis = pesquisarCliente(lc, nif);
244     if (pis == -1) {
245         System.out.println("Cliente não encontrado.");
246     }
247
248     int pos = pesquisarCliente(lc, nif);
249     if(pos != -1){
250         while(pos < lc.n_clientes -1){
251             lc.l_clientes[pos] = lc.l_clientes[pos+1];

```

```

252             pos++;
253         }
254         lc.l_clientes[lc.n_clientes -1] = null;
255         lc.n_clientes--;
256
257         if(lc.l_clientes.length - lc.n_clientes >= INCREMENTO
258             && lc.l_clientes.length > TAMANHO_INICIAL_CLIENTES){
259             Cliente[] temp = new Cliente[lc.l_clientes.length - INCREMENTO];
260             for(int i=0; i<temp.length;i++){
261                 temp[i] = lc.l_clientes[i];
262             }
263             lc.l_clientes = temp;
264             System.out.println("Cliente removido com sucesso!!!");
265             return true;
266         }
267         return false;
268     }
269
270 /**
271 * Função que imprime informações sobre os clientes armazenados na lista "clientes".
272 * @param clientes
273 */
274 static void imprimirClientes(ListaClientes clientes){
275     for(int i=0;i<clientes.n_clientes;i++){
276         System.out.println("ID: "+clientes.l_clientes[i].id);
277         System.out.println("NIF: "+clientes.l_clientes[i].NIF);
278         System.out.println("Nome: "+clientes.l_clientes[i].nome);
279         System.out.println("Morada: "+clientes.l_clientes[i].morada);
280         System.out.println("País: "+clientes.l_clientes[i].pais);
281         System.out.println("-----");
282     }
283 }
284
285 //-----
286 //----- Gestão de Produtos -----
287
288 /**
289 * Método, utilizado para ler informações de um arquivo chamado "produtos.txt"
290 * e adicioná-las a uma lista de produtos
291 * @param produtos
292 * @throws FileNotFoundException
293 */
294 static void LerFicheiroProdutos(ListaProdutos produtos) throws FileNotFoundException{
295     File f = new File("produtos.txt");
296     Scanner s = new Scanner(f);
297     Produto temp = new Produto();
298     while(s.hasNextLine()){
299         String linha = s.nextLine();
300         String[] valores = linha.split(";");
301         if(valores[0].equals("PRODUTO")){
302             temp = new Produto();

```

```

303     temp.codigo = Integer.parseInt(valores[1]);
304     temp.nome = valores[2];
305     temp.dimensoes = valores[3];
306     temp.preco = Integer.parseInt(valores[4]);
307     temp.tipo = valores[5];
308     temp.quantidade = Integer.parseInt(valores[6]);
309     temp.estado = valores[7];
310     adicionarProduto(produtos, temp);
311     nProduto = temp.codigo+1;
312 }else if(valores[0].equals("COMPONENTES")){
313     Componente comp = new Componente();
314     comp.cod_componente = Integer.parseInt(valores[1]);
315     comp.descricao = valores[2];
316     comp.quantidade = Integer.parseInt(valores[3]);
317     comp.unidade = valores[4];
318     adicionar_componente(produtos, temp.codigo, comp);
319     nComponente = comp.cod_componente+1;
320 }
321 }
322 s.close();
323 }
324 /**
325 * Método encontrado para inicializar para ser usado para
326 * gear um código ou id único
327 */
328 static int nProduto = 1;
329 /**
330 * Função que criará um ficheiro de texto "produtos.txt" e escreve nele
331 * informações sobre produtos e componentes.
332 * @param produtos
333 * @throws IOException
334 */
335 static void criarFicheiroProdutos(ListaProdutos produtos) throws IOException{
336     File f = new File("produtos.txt");
337     BufferedWriter wr = new BufferedWriter(new FileWriter(f));
338     for(int i=0;i<produtos.n_produtos;i++){
339         wr.write("PRODUTO;" +
340             +produtos.l_produtos[i].codigo+";" +
341             +produtos.l_produtos[i].nome+";" +
342             +produtos.l_produtos[i].dimensoes+";" +
343             +produtos.l_produtos[i].preco+";" +
344             +produtos.l_produtos[i].tipo+";" +
345             +produtos.l_produtos[i].quantidade+";" +
346             +produtos.l_produtos[i].estado+"\n");
347         for (int j = 0; j < produtos.l_produtos[i].n_componentes; j++) {
348             wr.write("COMPONENTES;" +
349                 +produtos.l_produtos[i].l_componentes[j].cod_componente + ";" +
350                 +produtos.l_produtos[i].l_componentes[j].descricao + ";" +
351                 +produtos.l_produtos[i].l_componentes[j].quantidade + ";"
352             )
353         }
354     }
355 }

```

```

354             + produtos.l_produtos[i].l_componentes[j].unidade + "\n");
355         }
356     }
357     wr.flush();
358     wr.close();
359 }
360
361 /**
362 * O método pesquisar é uma função que procura por um produto na
363 * com base em seu código
364 * @param produtos
365 * @param código
366 * @return
367 */
368 static int pesquisarProdutos(ListaProdutos produtos, int código) {
369     int pos = -1, i=0;
370     while(pos == -1 && i < produtos.n_produtos ){
371         if(produtos.l_produtos[i].código == código){
372             pos = i;
373         }
374         i++;
375     }
376     return pos;
377 }
378
379 /**
380 * Função responsável por adicionar um novo produto à lista.
381 * @param produtos
382 * @param novo
383 * @return
384 */
385 static boolean adicionarProduto(ListaProdutos produtos, Produto novo) {
386     int pos = pesquisarProdutos(produtos, novo.código);
387     if (pos != -1) {
388         return false;
389     }
390
391     if (produtos.n_produtos == produtos.l_produtos.length) {
392         if (produtos.n_produtos != TAMANHO_MAXIMO_PRODUTOS) {
393             Produto[] temp = new Produto[produtos.n_produtos + INCREMENTO];
394             for (int i = 0; i < produtos.n_produtos; i++) {
395                 temp[i] = produtos.l_produtos[i];
396             }
397             produtos.l_produtos = temp;
398         } else {
399             return false;
400         }
401     }
402     produtos.l_produtos[produtos.n_produtos++] = novo;
403     return true;
404 }

```

```

405
406 /**
407 * Função responsável por adicionar um novo componente à lista de produtos
408 * @param produtos
409 * @param codigo
410 * @param novo
411 * @return
412 */
413 static boolean adicionar_componente(ListaProdutos produtos, int codigo, Componente novo) {
414     int pos = pesquisarProdutos(produtos, codigo);
415     if (pos == -1) {
416         return false;
417     }
418     if (produtos.l_produtos[pos].l_componentes == null) {
419         produtos.l_produtos[pos].l_componentes = new Componente[10];
420     }
421     if (produtos.l_produtos[pos].n_componentes == produtos.l_produtos[pos].l_componentes.length) {
422
423         //aumentar a capacidade
424     }
425     produtos.l_produtos[pos].l_componentes[produtos.l_produtos[pos].n_componentes++] = novo;
426     return true;
427 }
428
429 /**
430 * Imprime as informações sobre um produto com base no seu código
431 * É feito por meio de um loop que percorre toda a lista
432 * @param produtos
433 * @param código
434 */
435 static void LocalizarProdutos(ListaProdutos produtos, int código){
436     for(int i = 0 ; i < produtos.n_produtos;i++){
437         if(produtos.l_produtos[i].codigo == código){
438             System.out.println("-----+");
439             System.out.println(" CÓDIGO: "+produtos.l_produtos[i].codigo);
440             System.out.println(" |-----| ");
441             System.out.println(" | Nome | "+produtos.l_produtos[i].nome);
442             System.out.println(" | Dimensões | "+produtos.l_produtos[i].dimensoes);
443             System.out.println(" | Preço | "+produtos.l_produtos[i].preco+" Euros");
444             System.out.println(" | Tipo | "+produtos.l_produtos[i].tipo);
445             System.out.println(" | Tipo | "+produtos.l_produtos[i].quantidade);
446             System.out.println(" | Estado | "+produtos.l_produtos[i].estado);
447             System.out.println("-----+");
448         }
449     }
450 }
451
452 /**
453 * Função que permite editar um produto existente numa lista d
454 * A função tem por entrada uma variável de Scanner e uma instância da
455 * classe ListaProdutos

```

```
456     * @param sc
457     * @param produtos
458     */
459     static void editarProdutos(Scanner sc, ListaProdutos produtos) {
460         System.out.print("Digite o Código do Produto a editar: ");
461         int codigo = sc.nextInt();
462         int pos = pesquisarProdutos(produtos, codigo);
463         if (pos == -1) {
464             System.out.println("Produto não encontrado.");
465             return;
466         }
467         Produto produto = produtos.l_produtos[pos];
468         int editar;
469         do {
470             System.out.println("+-----+");
471             System.out.println(" |      Editar Produto | ");
472             System.out.println("+-----+");
473             System.out.println(" | 1 | Nome          | ");
474             System.out.println(" | 2 | Dimensões       | ");
475             System.out.println(" | 3 | Preço           | ");
476             System.out.println(" | 4 | Tipo            | ");
477             System.out.println(" | 5 | Quantidade       | ");
478             System.out.println(" | 0 | MENU            | ");
479             System.out.println("+-----+");
480             System.out.print("Escolha uma das seguintes opções: ");
481             editar = sc.nextInt();
482             switch (editar) {
483                 case 1:
484                     System.out.print("Insirir nova descrição: ");
485                     sc.nextLine();
486                     produto.nome = sc.nextLine();
487                     break;
488                 case 2:
489                     System.out.print("Insirir a novas dimensões: ");
490                     sc.nextLine();
491                     produto.dimensoes = sc.nextLine();
492                     break;
493                 case 3:
494                     System.out.print("Insirir novo preço: ");
495                     sc.nextLine();
496                     produto.preco = sc.nextInt();
497                     break;
498                 case 4:
499                     System.out.print("Novo tipo de produto: ");
500                     sc.nextLine();
501                     produto.tipo = sc.nextLine();
502                     break;
503                 case 5:
504                     System.out.print("Editar Quantidade: ");
505                     sc.nextLine();
506                     produto.quantidade = sc.nextInt();
```

```

507             break;
508         case 0:
509             break;
510         default:
511             System.out.println("Opção inválida.");
512     }
513 } while (editar != 0);
514 }
515
516 /**
517 * Função remove um cliente da lista de produtos em função do código
518 * @param produtos
519 * @param código
520 * @return
521 */
522 static boolean removerProdutos(ListaProdutos produtos, int código){
523     int pes = pesquisarProdutos(produtos, código);
524     if (pes == -1) {
525         System.out.println("Produto não existe!.");
526     }
527
528     int pos = pesquisarProdutos(produtos, código);
529     if(pos != -1){
530         while(pos <produtos.n_produtos -1){
531             produtos.l_produtos[pos] = produtos.l_produtos[pos+1];
532             pos++;
533         }
534         produtos.l_produtos[produtos.n_produtos -1] = null;
535         produtos.n_produtos--;
536
537         if(produtos.l_produtos.length - produtos.n_produtos >= INCREMENTO &&
538             produtos.l_produtos.length > TAMANHO_INICIAL_PRODUTOS){
539             Produto[] temp = new Produto[produtos.l_produtos.length -INCREMENTO];
540             for(int i=0; i<temp.length;i++){
541                 temp[i] = produtos.l_produtos[i];
542             }
543             produtos.l_produtos = temp;
544         }System.out.println("Produto removido com sucesso!!!");
545         return true;
546     }
547     return false;
548 }
549
550 /**
551 * Função que imprime informações sobre os produtos armazenados na lista
552 * @param produtos
553 */
554 static void imprimirProdutos(ListaProdutos produtos){
555     for(int i=0;i<produtos.n_produtos;i++){
556         System.out.println("-----+");
557         System.out.println("    CÓDIGO: " + produtos.l_produtos[i].codigo);

```

```

558     System.out.println(" | -----+-----| ");
559     System.out.println(" | Nome      | "+produtos.l_produtos[i].nome);
560     System.out.println(" | Dimensões | "+produtos.l_produtos[i].dimensoes +" cm");
561     System.out.println(" | Preço     | "+produtos.l_produtos[i].preco+" Euros");
562     System.out.println(" | Tipo      | "+produtos.l_produtos[i].tipo);
563     System.out.println(" | Quantidade | "+produtos.l_produtos[i].quantidade+" Unidades");
564     System.out.println(" | Estado    | "+produtos.l_produtos[i].estado);
565     System.out.println(" +-----+-----+");
566   }
567 }
568
569 //-----+
570 //----- COMPONENTES -----+
571
572 /**
573 *
574 * @param componentes
575 * @throws FileNotFoundException
576 */
577 static void LerFicheiroComponentes(ListaComponentes componentes) throws FileNotFoundException{
578   File f = new File("componentes.txt");
579   Scanner s = new Scanner(f);
580   Componente temp = new Componente();
581   while(s.hasNextLine()){
582     String linha = s.nextLine();
583     String[] valores = linha.split(";");
584     if(valores[0].equals("COMPONENTE")){
585       temp = new Componente();
586       temp.cod_componente = Integer.parseInt(valores[1]);
587       temp.descricao = valores[2];
588       temp.quantidade = Integer.parseInt(valores[3]);
589       temp.unidade = valores[4];
590       adicionarComponentes(componentes, temp);
591       nComponente = temp.cod_componente+1;
592     }
593   }
594 }
595
596 /**
597 * Forma encontrada para gerar um código ou id único
598 */
599 static int nComponente = 1;
600
601 /**
602 * Função que criará um ficheiro de texto "componentes.txt" e escreve nele
603 * informações sobre componentes.
604 * @param componentes
605 * @throws IOException
606 */
607 static void criarFicheiroComponentes(ListaComponentes componentes) throws IOException{
608   File f = new File("componentes.txt");

```

```

609     BufferedWriter wr = new BufferedWriter(new FileWriter(f));
610     for(int i=0;i<componentes.n_componentes;i++){
611         wr.write("COMPONENTE;" +
612                 +componentes.l_componentes[i].cod_componente+";" +
613                 +componentes.l_componentes[i].descricao+";" +
614                 +componentes.l_componentes[i].quantidade+";" +
615                 +componentes.l_componentes[i].unidade+"\n");
616     }
617     wr.flush();
618     wr.close();
619 }
620
621 /**
622 * O método pesquisar é uma função que procura por componentes na
623 * com base em seu código
624 * @param componentes
625 * @param código
626 * @return
627 */
628 static int pesquisarComponentes(ListaComponentes componentes, int código) {
629     int pos = -1,i=0;
630     while(pos == -1 && i < componentes.n_componentes ){
631         if(componentes.l_componentes[i].cod_componente == código){
632             pos = i;
633         }
634         i++;
635     }
636     return pos;
637 }
638
639 /**
640 * Função responsável por adicionar um novo produto à lista.
641 * @param componentes
642 * @param novo
643 * @return
644 */
645 static boolean adicionarComponentes(ListaComponentes componentes, Componente novo) {
646     int pos = pesquisarComponentes(componentes, novo.cod_componente);
647     if (pos != -1) {
648         return false;
649     }
650
651     if (componentes.n_componentes == componentes.l_componentes.length) {
652         if (componentes.n_componentes != TAMANHO_MAXIMO_COMPONENTES) {
653             Componente[] temp = new Componente[componentes.n_componentes + INCREMENTO];
654             for (int i = 0; i < componentes.n_componentes; i++) {
655                 temp[i] = componentes.l_componentes[i];
656             }
657             componentes.l_componentes = temp;
658         } else {
659             return false;

```

```

660         }
661     }
662     componentes.l_componentes[componentes.n_componentes++] = novo;
663     return true;
664 }
665 /**
666 * Método, utilizado para ler informações de um arquivo chamado
667 * "componentes.txt" e adicioná-las a uma lista
668 * @param componentes
669 * @param codigo
670 */
671 static void LocalizarComponentes(ListaComponentes componentes, int codigo){
672     for(int i = 0 ; i < componentes.n_componentes;i++ ){
673         if(componentes.l_componentes[i].cod_componente == codigo){
674             System.out.println("+"-----+");
675             System.out.println(" CÓDIGO: "+ componentes.l_componentes[i].cod_componente);
676             System.out.println(" |-----| ");
677             System.out.println(" | Descrição | "+componentes.l_componentes[i].descricao);
678             System.out.println(" | Quantidade | "+componentes.l_componentes[i].quantidade);
679             System.out.println(" | Unidade | "+componentes.l_componentes[i].unidade);
680             System.out.println("+"-----+");
681         }
682     }
683 }
684 }
685 /**
686 * Função que permite editar componentes existentes numa lista
687 * A função tem por entrada uma variável de Scanner e uma instância da
688 * lista
689 * @param sc
690 * @param componentes
691 */
692 static void editarComponentes(Scanner sc, ListaComponentes componentes) {
693     System.out.print("Digite o Código do Componente a editar: ");
694     int codigo = sc.nextInt();
695     int pos = pesquisarComponentes(componentes, codigo);
696     if (pos == -1) {
697         System.out.println("Componente não encontrado.");
698         return;
699     }
700     Componente componente = componentes.l_componentes[pos];
701     int editar;
702     do {
703         System.out.println("+"-----+");
704         System.out.println(" | Editar Componente | ");
705         System.out.println("+"-----+");
706         System.out.println(" | 1 | Descrição | ");
707         System.out.println(" | 2 | Quantidade | ");
708         System.out.println(" | 3 | Unidade | ");
709         System.out.println(" | 0 | MENU | ");
710     }

```

```

711     System.out.println("-----+");
712     System.out.print("Escolha uma das seguintes opções: ");
713     editar = sc.nextInt();
714     switch (editar) {
715         case 1:
716             System.out.print("Inserir nova descrição: ");
717             sc.nextLine();
718             componente.descricao = sc.nextLine();
719             break;
720         case 2:
721             System.out.print("Inserir a nova quantidade: ");
722             sc.nextLine();
723             componente.quantidade = sc.nextInt();
724             break;
725         case 3:
726             System.out.print("Inserir nova unidade: ");
727             sc.nextLine();
728             componente.unidade = sc.nextLine();
729             break;
730         case 0:
731             break;
732         default:
733             System.out.println("Opção inválida.");
734     }
735     } while (editar != 0);
736 }
737
738 /**
739 * Função remove um cliente da lista de componentes em função do código
740 * @param componentes
741 * @param código
742 * @return
743 */
744 static boolean removerComponentes(ListaComponentes componentes, int código){
745     int pes = pesquisarComponentes(componentes, código);
746     if (pes == -1) {
747         System.out.println("Componente não existe!.");
748     }
749
750     int pos = pesquisarComponentes(componentes, código);
751     if(pos != -1){
752         while(pos < componentes.n_componentes -1){
753             componentes.l_componentes[pos] = componentes.l_componentes[pos+1];
754             pos++;
755         }
756         componentes.l_componentes[componentes.n_componentes -1] = null;
757         componentes.n_componentes--;
758
759         if(componentes.l_componentes.length - componentes.n_componentes >= INCREMENTO &&
760             componentes.l_componentes.length > TAMANHO_INICIAL_COMPONENTES){
761             Componente[] temp = new Componente[componentes.l_componentes.length -INCREMENTO];

```

```

762         for(int i=0; i<temp.length;i++){
763             temp[i] = componentes.l_componentes[i];
764         }
765         componentes.l_componentes = temp;
766     }System.out.println("Componente removido com sucesso!!!");
767     return true;
768 }
769 return false;
770 /**
771 * Função que imprime informações sobre os produtos armazenados na lista
772 * @param componentes
773 */
774 static void imprimirComponentes(ListaComponentes componentes){
775     for(int i=0;i<componentes.n_componentes;i++){
776         System.out.println("-----+");
777         System.out.println(" CÓDIGO: "+ componentes.l_componentes[i].cod_componente);
778         System.out.println("-----|");
779         System.out.println(" | Descrição | "+componentes.l_componentes[i].descricao);
780         System.out.println(" | Quantidade | "+componentes.l_componentes[i].quantidade);
781         System.out.println(" | Unidade | "+componentes.l_componentes[i].unidade);
782         System.out.println("-----+");
783     }
784 }
785 }
786 //-----
787 //----- PERFIL CLIENTE -----
788 /**
789 *
790 * @param encomendas
791 * @throws FileNotFoundException
792 */
793 static void LerFicheiroEncomendas(ListaEncomendas encomendas) throws FileNotFoundException {
794     File f = new File("encomendas.txt");
795     Scanner s = new Scanner(f);
796     Encomenda temp = new Encomenda();
797     while (s.hasNextLine()) {
798         String linha = s.nextLine();
799         String[] valores = linha.split(";");
800         if (valores[0].equals("ENCOMENDA")) {
801             temp = new Encomenda();
802             temp.id_encomenda = Integer.parseInt(valores[1]);
803             temp.nif_cliente = Integer.parseInt(valores[2]);
804             temp.preco= Integer.parseInt(valores[3]);
805             adicionarEncomenda(encomendas, temp);
806             nEncomenda = temp.id_encomenda+1;
807
808         } else {
809             if (valores[0].equals("PRODUTO")) {
810                 Produto produto = new Produto();

```

```

813         produto.nome = valores[1];
814         produto.preco = Integer.parseInt(valores[2]);
815         produto.quantidade = Integer.parseInt(valores[3]);
816         adicionarEncomenda(encomendas, temp);
817         nEncomenda = temp.id_encomenda + 1;
818     }
819 }
820 }
821 }
822 /**
823 *
824 * @param encomendas
825 * @param novo
826 * @return
827 */
828 static boolean adicionarEncomenda(ListaEncomendas encomendas, Encomenda novo) {
829     int pos = pesquisarEncomendas(encomendas, novo.id_encomenda);
830     if (pos != -1) {
831         return false;
832     }
833
834     if (encomendas.n_encomendas == encomendas.l_encomendas.length) {
835         if (encomendas.n_encomendas != TAMANHO_MAXIMO_ENCOMENDAS) {
836             Encomenda[] temp = new Encomenda[encomendas.n_encomendas + INCREMENTO];
837             for (int i = 0; i < encomendas.n_encomendas; i++) {
838                 temp[i] = encomendas.l_encomendas[i];
839             }
840             encomendas.l_encomendas = temp;
841         } else {
842             return false;
843         }
844     }
845     encomendas.l_encomendas[encomendas.n_encomendas++] = novo;
846     return true;
847 }
848 /**
849 *
850 * @param encomendas
851 * @param id
852 * @return
853 */
854 static int pesquisarEncomendas(ListaEncomendas encomendas, int id) {
855     int pos = -1, i=0;
856     while(pos == -1 && i < encomendas.n_encomendas ) {
857         if(encomendas.l_encomendas[i].id_encomenda == id){
858             pos = i;
859         }
860         i++;
861     }
862 }
863 }
```

```

864        return pos;
865    }
866
867    /**
868     * Método encontrado para inicializar para ser usado para
869     * gear um código ou id único
870     */
871    static int nEncomenda = 1;
872
873    /**
874     *
875     * @param encomendas
876     * @param id
877     * @param produtos
878     * @return
879     */
880    static boolean adicionarProdutosEncomenda(ListaEncomendas encomendas, int id, Produto produtos) {
881        int pos = pesquisarEncomendas(encomendas, id);
882        if (pos == -1) {
883            return false;
884        }
885        if (encomendas.l_encomendas[pos].l_produtos == null) {
886            encomendas.l_encomendas[pos].l_produtos = new Produto[10];
887        }
888        if (encomendas.l_encomendas[pos].n_produtos == encomendas.l_encomendas[pos].l_produtos.length) {
889        }
890        encomendas.l_encomendas[pos].l_produtos[encomendas.l_encomendas[pos].n_produtos++] = produtos;
891        return true;
892    }
893
894    /**
895     *
896     * @param encomendas
897     * @throws IOException
898     */
899    static void criarFicheiroEncomenda(ListaEncomendas encomendas) throws IOException {
900        File f = new File("encomendas.txt");
901        BufferedWriter wr = new BufferedWriter(new FileWriter(f, true));
902        for (int i = 0; i < encomendas.n_encomendas; i++) {
903            wr.write("ENCOMENDA:"
904                    + encomendas.l_encomendas[i].id_encomenda + ";"
905                    + encomendas.l_encomendas[i].nif_cliente + ";"
906                    + encomendas.l_encomendas[i].preco + "\n");
907            for (int j = 0; j < encomendas.l_encomendas[i].n_produtos; j++) {
908                wr.write("PRODUTO:" + encomendas.l_encomendas[i].l_produtos[j].nome + ";"
909                        + encomendas.l_encomendas[i].l_produtos[j].preco + ";"
910                        + encomendas.l_encomendas[i].l_produtos[j].quantidade + "\n");
911            }
912        }
913        wr.flush();
914        wr.close();

```

```

915     }
916
917     /**
918      *
919      * @param sc
920      * @param encomendas
921      */
922     static void editarEncomendas(Scanner sc, ListaEncomendas encomendas) {
923         System.out.print("Inserir o ID da encomenda a editar: ");
924         int id = sc.nextInt();
925         int pos = pesquisarEncomendas(encomendas, id);
926         if (pos == -1) {
927             System.out.println("ID não encontrado.");
928             return;
929         }
930         Encomenda encomenda = encomendas.l_encomendas[pos];
931         int editar;
932         do {
933             System.out.println("+-----+");
934             System.out.println("| Editar Encomenda |");
935             System.out.println("+-----+");
936             System.out.println("| 1 | NIF Cliente |");
937             System.out.println("| 2 | ID Produto |");
938             System.out.println("| 3 | Preço |");
939             System.out.println("| 0 | MENU |");
940             System.out.println("+-----+");
941             System.out.print("Escolha uma das seguintes opções: ");
942             editar = sc.nextInt();
943             switch (editar) {
944                 case 1:
945                     System.out.print("Inserir novo NIF: ");
946                     sc.nextLine();
947                     encomenda.nif_cliente = sc.nextInt();
948                     break;
949                 case 2:
950                     System.out.print("Inserir novo ID de Produto: ");
951                     sc.nextLine();
952                     encomenda.l_produtos[2].codigo = sc.nextInt();
953                     break;
954                 case 3:
955                     System.out.print("Inserir novo Preço: ");
956                     sc.nextLine();
957                     encomenda.l_produtos[3].preco = sc.nextInt();
958                     break;
959                 case 0:
960                     System.out.println("MENU");
961                     break;
962                 default:
963                     System.out.println("Opção inválida.");
964             }
965         } while (editar != 0);

```

```

966     }
967
968     /**
969      *
970      * @param encomendas
971      * @param id
972      * @return
973      */
974     static boolean removerEncomendas(ListaEncomendas encomendas, int id){
975         int pes = pesquisarEncomendas(encomendas, id);
976         if (pes == -1) {
977             System.out.println("Componente não existe!.");
978         }
979
980         int pos = pesquisarEncomendas(encomendas, id);
981         if(pos != -1){
982             while(pos <encomendas.n_encomendas -1){
983                 encomendas.l_encomendas[pos] = encomendas.l_encomendas[pos+1];
984                 pos++;
985             }
986             encomendas.l_encomendas[encomendas.n_encomendas -1] = null;
987             encomendas.n_encomendas--;
988
989             if(encomendas.l_encomendas.length - encomendas.n_encomendas >= INCREMENTO &&
990                 encomendas.l_encomendas.length > TAMANHO_INICIAL_ENCOMENDAS){
991                 Encomenda[] temp = new Encomenda[encomendas.l_encomendas.length -INCREMENTO];
992                 for(int i=0; i<temp.length;i++){
993                     temp[i] = encomendas.l_encomendas[i];
994                 }
995                 encomendas.l_encomendas = temp;
996             }System.out.println("Encomenda eliminada com sucesso!!!");
```

Encomenda eliminada com sucesso!!!

```

997             return true;
998         }
999         return false;
1000     }
1001
1002     /**
1003      *
1004      * @param encomendas
1005      */
1006     static void imprimirEncomendas(ListaEncomendas encomendas){
1007         for(int i=0;i<encomendas.n_encomendas;i++){
1008             System.out.println("+-----+");
1009             System.out.println("    ENCOMENDA ID: " + encomendas.l_encomendas[i].id_encomenda);
1010             System.out.println("    |-----|");
1011             System.out.println("    | ID:          | " +encomendas.l_encomendas[i].id_encomenda);
1012             System.out.println("    | NIF Cliente | " +encomendas.l_encomendas[i].nif_cliente);
1013             System.out.println("    | Preço TOTAL (€) | " +encomendas.l_encomendas[i].preco);
1014             System.out.println("+-----+");
1015         }
1016     }

```

```

1017
1018 /**
1019 * @param args the command line arguments
1020 * @throws java.io.IOException
1021 */
1022 public static void main(String[] args) throws IOException{
1023 /**
1024 * Criamos uma instância da classe "ListaClientes" e inicializamos
1025 * os atributos "l_clientes" com um array de tamanho "TAMANHO_INICIAL_CLIENTES".
1026 * O atributo "n_clientes" é inicializado com 0.
1027 * Em seguida, a função "lerFicheiroClientes" é chamada passando
1028 * "listaclientes" como parâmetro. Essa função irá ler um arquivo de
1029 * texto ou outra fonte de dados para preencher a lista de clientes.
1030 *
1031 */
1032
1033 ListaClientes listaclientes = new ListaClientes();
1034 listaclientes.l_clientes = new Cliente[TAMANHO_INICIAL_CLIENTES];
1035 listaclientes.n_clientes = 0;
1036 LerFicheiroClientes(listaclientes);
1037
1038 ListaProdutos listaprodutos = new ListaProdutos();
1039 listaprodutos.l_produtos = new Produto[TAMANHO_INICIAL_PRODUTOS];
1040 listaprodutos.n_produtos = 0;
1041 LerFicheiroProdutos(listaprodutos);
1042
1043 ListaComponentes listacomponentes = new ListaComponentes();
1044 listacomponentes.l_componentes = new Componente[TAMANHO_INICIAL_COMPONENTES];
1045 listacomponentes.n_componentes = 0;
1046 LerFicheiroComponentes(listacomponentes);
1047
1048 ListaEncomendas listaencomendas= new ListaEncomendas();
1049 listaencomendas.l_encomendas = new Encomenda[TAMANHO_INICIAL_ENCOMENDAS];
1050 listaencomendas.n_encomendas= 0;
1051 LerFicheiroEncomendas(listaencomendas);
1052
1053 Scanner sc = new Scanner(System.in);
1054 int menu;
1055 do {
1056     System.out.println("");
1057     System.out.println(" _ | = | _____");
1058     System.out.println(" / \\" );
1059     System.out.println(" / \\" );
1060     System.out.println(" / \\" );
1061     System.out.println(" [ ] /--\\" [ ] [ ]");
1062     System.out.println(" [ ] [ ] . [ ] [ ] [ ]");
1063     System.out.println(" ( ) [ ] [ ] [ ] [ ] ( )");
1064     System.out.println(" ( ) - [ ] - [ ] == [ ] - [ ] - [ ( ) ]");
1065     System.out.println("");
1066     System.out.println("BEM VINDOS À LOJA RADIO CONTROL\n");
1067     System.out.println("+-----+");

```

```

1068     System.out.println("          MENU      ");
1069     System.out.println("+-----+");
1070     System.out.println(" | 1 | Perfil Administrador | ");
1071     System.out.println(" | 2 | Perfil Cliente   | ");
1072     System.out.println(" | 0 | SAIR             | ");
1073     System.out.println("+-----+");
1074     System.out.print("Escolha uma das seguintes opções: ");
1075     menu = sc.nextInt();
1076     System.out.println("\n");
1077     switch (menu) {
1078         case 1:
1079             int admin;
1080             do {
1081                 System.out.println("+-----+");
1082                 System.out.println(" |          ADMINISTRADOR      | ");
1083                 System.out.println(" |-----| ");
1084                 System.out.println(" | 1 | Gestão de Clientes    | ");
1085                 System.out.println(" | 2 | Gestão de Produtos    | ");
1086                 System.out.println(" | 3 | Gestão de Componentes | ");
1087                 System.out.println(" | 4 | Registo de Encomendas | ");
1088                 System.out.println(" | 0 | MENU                  | ");
1089                 System.out.println("+-----+");
1090                 System.out.print("Escolha uma das seguintes opções: ");
1091                 admin = sc.nextInt();
1092                 System.out.println("\n");
1093                 switch (admin) {
1094                     case 1:
1095                         int menuclientes;
1096                         do {
1097                             System.out.println("+-----+");
1098                             System.out.println(" |          GESTÃO CLIENTES    | ");
1099                             System.out.println(" |-----| ");
1100                             System.out.println(" | 1 | Adicionar Clientes    | ");
1101                             System.out.println(" | 2 | Pesquisar Clientes    | ");
1102                             System.out.println(" | 3 | Editar Clientes       | ");
1103                             System.out.println(" | 4 | Remover Clientes      | ");
1104                             System.out.println(" | 5 | IMPRIMIR Lista Clientes | ");
1105                             System.out.println(" | 6 | GRAVAR                | ");
1106                             System.out.println(" | 0 | MENU                  | ");
1107                             System.out.println("+-----+");
1108                             System.out.print("Escolha uma das seguintes opções: ");
1109                             menuclientes = sc.nextInt();
1110                             System.out.println("\n");
1111                             sc.nextLine();
1112                             switch (menuclientes) {
1113                                 case 1:
1114                                     try {
1115                                         System.out.println("----- ADICIONAR CLIENTES -----");
1116                                         Cliente temp = new Cliente();
1117                                         temp.id = nCliente++;
1118                                         boolean nifExiste = true;

```

```

1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169

        while (nifExiste) {
            System.out.println("ID: "+temp.id);
            System.out.print("NIF: ");
            try {
                temp.NIF = Integer.parseInt(sc.nextLine());
                nifExiste = verificarNIF(listaclientes, temp.NIF);
                if(nifExiste) {
                    System.out.println("NIF já existente. Por favor insira outro.");
                }
            } catch (NumberFormatException e) {
                System.out.println("Insira apenas números para o NIF.");
            }
        }
        System.out.print("Nome: ");
        temp.nome = sc.nextLine();
        System.out.print("Morada: ");
        temp.morada = sc.nextLine();
        System.out.print("País: ");
        temp.pais = sc.nextLine();
        if (adicionarCliente(listaclientes, temp)){
            System.out.println("Cliente adicionado com sucesso!");
        }
    } catch (Exception e) {
    }
    break;
case 2:
    System.out.println("----- PESQUISAR CLIENTES ----- ");
    System.out.print("Inserir o NIF do cliente: ");
    int nif = sc.nextInt();
    sc.nextLine();
    LocalizarCliente(listaclientes, nif);
    try {
        System.out.println("Pressione 'Enter' para voltar ao MENU");
        System.in.read();
    } catch (IOException e) {
    }
    break;
case 3:
    System.out.println("----- EDITAR CLIENTES ----- ");
    editarClientes(sc, listaclientes);
    break;
case 4:
    System.out.println("----- REMOVER CLIENTES -----");
    System.out.print("Inserir NIF do Cliente: ");
    int idRemover = sc.nextInt();
    removerCliente(listaclientes, idRemover);
    break;
case 5:
    System.out.println("----- LISTA DE CLIENTES ----- ");
    imprimirClientes(listaclientes);
    break;
}

```

```

1170
1171     case 6:
1172         System.out.println("----- GRAVAR FICHEIRO -----");
1173         criarFicheiroClientes(listaclientes);
1174         System.out.println("As alterações foram gravadas com sucesso!!!");
1175         break;
1176     case 0:
1177         System.out.println("----- SAIR -----");
1178         System.out.println("MENU");
1179         break;
1180     default:
1181         System.out.println("Opção inválida!");
1182     }
1183 } while (menuclientes !=0);
1184 break;
case 2:
1185     int menuprodutos;
1186     do {
1187         System.out.println("+-----+");
1188         System.out.println(" | PRODUTOS |");
1189         System.out.println(" |-----|");
1190         System.out.println(" | 1 | Adicionar Produtos |");
1191         System.out.println(" | 2 | Pesquisar Produtos |");
1192         System.out.println(" | 3 | Editar Produtos |");
1193         System.out.println(" | 4 | Remover Produtos |");
1194         System.out.println(" | 5 | IMPRIMIR Lista Produtos |");
1195         System.out.println(" | 6 | GRAVAR |");
1196         System.out.println(" | 0 | MENU |");
1197         System.out.println("+-----+");
1198         System.out.print("Escolha uma das seguintes opções: ");
1199         menuprodutos = sc.nextInt();
1200         System.out.println("\n");
1201         switch (menuprodutos) {
1202             case 1:
1203
1204                 System.out.println("----- ADICIONAR PRODUTOS -----");
1205                 Produto temp = new Produto();
1206                 temp.codigo = nProduto++;
1207                 sc.nextLine();
1208                 System.out.print("ID: "+temp.codigo);
1209                 System.out.print("Nome: ");
1210                 temp.nome = sc.nextLine();
1211                 System.out.print("Dimensões(CxLxA): ");
1212                 temp.dimensoes = sc.nextLine();
1213                 System.out.print("Preço: ");
1214                 temp.preco = sc.nextInt();
1215                 sc.nextLine();
1216                 System.out.print("Tipo: ");
1217                 temp.tipo = sc.nextLine();
1218                 System.out.print("Quantidade: ");
1219                 temp.quantidade = sc.nextInt();
1220                 System.out.println("");

```

```

1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271

        System.out.print("Deseja inserir componentes? (1 - sim, 0 - não): ");
        menuprodutos = sc.nextInt();
        sc.nextLine();
        Componente comp = new Componente();
        int menucomponentes = 1;
        try{
            while (menucomponentes == 1) {
                System.out.println("----- ADICIONAR COMPONENTES -----");
                comp.cod_componente = nComponente++;
                System.out.println("ID: "+comp.cod_componente);
                System.out.print("Descrição: ");
                comp.descricao = sc.nextLine();
                System.out.print("Quantidade: ");
                comp.quantidade = sc.nextInt();
                sc.nextLine();
                System.out.print("Unidade: ");
                comp.unidade = sc.nextLine();
                System.out.println("Componente adicionado com sucesso!\n");
                System.out.print("Deseja inserir mais componentes? (1 - sim, 0 - não): ");
                menucomponentes = sc.nextInt();
                sc.nextLine();
                if (adicionar_componente(listaprodutos, menucomponentes, comp)){
                    System.out.println("Componente adicionado com sucesso!");
                }
            }
        } catch (Exception e) {
        }
        System.out.println("");
        if (adicionarProduto(listaprodutos, temp)){
            System.out.println("Produto adicionado com sucesso!");
        }
        break;
    case 2:
        System.out.println("----- PESQUISAR PRODUTOS ----- ");
        System.out.print("Inserir o codigo do produto: ");
        int codigo = sc.nextInt();
        sc.nextLine();
        LocalizarProdutos(listaprodutos, codigo);
        try {
            System.out.println("Pressione 'Enter' para voltar ao MENU");
            System.in.read();
        } catch (IOException e) {
        }
        break;
    case 3:
        System.out.println("----- EDITAR PRODUTOS ----- ");
        editarProdutos(sc, listaprodutos);
        break;
    case 4:
        System.out.println("----- REMOVER PRODUTOS ----- ");
        System.out.print("Inserir Código do Produto: ");

```

```

1272             int codigoRemover = sc.nextInt();
1273             removerProdutos(listaprodutos, codigoRemover);
1274             break;
1275         case 5:
1276             System.out.println("----- LISTA DE PRODUTOS ----- ");
1277             imprimirProdutos(listaprodutos);
1278             break;
1279         case 6:
1280             System.out.println("----- GRAVAR FICHEIRO -----");
1281             criarFicheiroProdutos(listaprodutos);
1282             System.out.println("As alterações foram gravadas com sucesso!!!");
1283             break;
1284         case 0:
1285             System.out.println("MENU");
1286             break;
1287         default:
1288             System.out.println("Opção inválida!");
1289         }
1290     } while (menuprodutos !=0);
1291     break;
1292 case 3:
1293     int menucomponentes;
1294     do {
1295         System.out.println("+-----+");
1296         System.out.println(" |          COMPONENTES          |");
1297         System.out.println(" |-----|");
1298         System.out.println(" | 1 | Adicionar Componentes   |");
1299         System.out.println(" | 2 | Pesquisar Componentes   |");
1300         System.out.println(" | 3 | Editar Componentes      |");
1301         System.out.println(" | 4 | Remover Componentes     |");
1302         System.out.println(" | 5 | IMPRIMIR Lista Componentes |");
1303         System.out.println(" | 6 | GRAVAR                  |");
1304         System.out.println(" | 0 | MENU                    |");
1305         System.out.println("+-----+");
1306         System.out.print("Escolha uma das seguintes opções: ");
1307         menucomponentes = sc.nextInt();
1308         System.out.println("\n");
1309         switch (menucomponentes) {
1310             case 1:
1311                 try {
1312                     System.out.println("----- ADICIONAR COMPONENTES -----");
1313                     Componente temp = new Componente();
1314                     temp.cod_componente = nComponente++;
1315                     sc.nextLine();
1316                     System.out.print("ID: "+temp.cod_componente);
1317                     System.out.print("Descrição: ");
1318                     temp.descricao = sc.nextLine();
1319                     System.out.print("Quantidade: ");
1320                     temp.quantidade = sc.nextInt();
1321                     sc.nextLine();
1322                     System.out.print("Unidade: ");

```

```

1323         temp.unidade = sc.nextLine();
1324         System.out.println("");
1325         if (adicionarComponentes(listacomponentes, temp)){
1326             System.out.println("Componente adicionado com sucesso!");
1327         }
1328     } catch (Exception e) {
1329     }
1330     break;
1331 case 2:
1332     System.out.println("----- PESQUISAR COMPONENTES ----- ");
1333     System.out.print("Inserir o codigo do componente: ");
1334     int codigo = sc.nextInt();
1335     sc.nextLine();
1336     LocalizarComponentes(listacomponentes, codigo);
1337     try {
1338         System.out.println("Pressione 'Enter' para voltar ao MENU");
1339         System.in.read();
1340     } catch (IOException e) {
1341     }
1342     break;
1343 case 3:
1344     System.out.println("----- EDITAR COMPONENTES ----- ");
1345     editarComponentes(sc, listacomponentes);
1346     break;
1347 case 4:
1348     System.out.println("----- REMOVER COMPONENTES -----");
1349     System.out.print("Inserir Código do Componente: ");
1350     int codigoRemover = sc.nextInt();
1351     removerComponentes(listacomponentes, codigoRemover);
1352     break;
1353 case 5:
1354     System.out.println("----- LISTA DE COMPONENTES ----- ");
1355     imprimirComponentes(listacomponentes);
1356     break;
1357 case 6:
1358     System.out.println("----- GRAVAR FICHEIRO -----");
1359     criarFicheiroComponentes(listacomponentes);
1360     System.out.println("As alterações foram gravadas com sucesso!!!");
1361     break;
1362 case 0:
1363     System.out.println("MENU");
1364     break;
1365 default:
1366     System.out.println("Opção inválida!");
1367 }
1368 } while (menucomponentes !=0);
1369
1370     break;
1371 case 4:
1372     int menuencomendas;
1373     do {

```

```

1374
1375     System.out.println("-----+-----+");
1376     System.out.println(" | ENCOMENDAS | ");
1377     System.out.println(" |-----|-----|");
1378     System.out.println(" | 1 | Registar Encomendas | ");
1379     System.out.println(" | 2 | Editar Encomendas | ");
1380     System.out.println(" | 3 | Eliminar Encomendas | ");
1381     System.out.println(" | 4 | Lista de Encomendas | ");
1382     System.out.println(" | 0 | MENU | ");
1383     System.out.println("-----+-----+");
1384     System.out.print("Escolha uma das seguintes opções: ");
1385     menuencomendas = sc.nextInt();
1386     System.out.println("\n");
1387     switch (menuencomendas) {
1388         case 1:
1389             System.out.println("----- REGISTAR ENCOMENDAS ----- ");
1390             try {
1391                 Encomenda temp = new Encomenda();
1392                 Produto produto = new Produto();
1393                 temp.id_encomenda = nEncomenda++;
1394                 System.out.print("Insira o NIF do cliente: ");
1395                 temp.nif_cliente = sc.nextInt();
1396                 sc.nextLine();
1397                 adicionarEncomenda(listaencomendas, temp);
1398                 int pos = pesquisarCliente(listaclientes, temp.nif_cliente);
1399                 if (pos != -1) {
1400                     int opcao = 1;
1401                     while (opcao == 1) {
1402                         System.out.print("Insira o ID Produto: ");
1403                         produto.codigo = sc.nextInt();
1404                         System.out.print("Inserir a quantidade pretendida: ");
1405                         produto.quantidade = sc.nextInt();
1406                         adicionarProdutosEncomenda(listaencomendas, pos, produto);
1407                         System.out.println("Encomenda adicionada com sucesso!\n");
1408                         System.out.print("Pretende inserir mais produtos? (1 - sim, 0 - não): ");
1409                         opcao = sc.nextInt();
1410                         sc.nextLine();
1411                     }
1412                 } else {
1413                     System.out.println("O NIF introduzido não está registado no sistema");
1414                 }
1415             } catch (Exception e) {
1416             }
1417             criarFicheiroEncomenda(listaencomendas);
1418             break;
1419         case 2:
1420             System.out.println("----- EDITAR ENCOMENDAS -----");
1421             editarEncomendas(sc, listaencomendas);
1422             break;
1423         case 3:
1424             System.out.println("----- ELIMINAR ENCOMENDAS -----");
1425             System.out.print("Inserir ID da Encomenda: ");

```

```

1425             int idRemover = sc.nextInt();
1426             removerEncomendas(listaencomendas, idRemover);
1427             break;
1428         case 4:
1429             System.out.println("----- LISTA ENCOMENDAS -----");
1430             imprimirEncomendas(listaencomendas);
1431             break;
1432         case 0:
1433             System.out.println("MENU");
1434             break;
1435         default:
1436             System.out.println("Opção inválida!");
1437         }
1438     }while (menuencomendas !=01);
1439     break;
1440     case 0:
1441         System.out.println("MENU");
1442         break;
1443     default:
1444         System.out.println("Opção inválida!");
1445     }
1446 }while (admin!=0);
1447 break;
1448 case 2:
1449     int menuCliente;
1450     do {
1451         System.out.println("+-----+");
1452         System.out.println(" | ENCOMENDAS |");
1453         System.out.println(" |-----|");
1454         System.out.println(" | 1 | Registar Encomendas |");
1455         System.out.println(" | 2 | Editar Encomendas |");
1456         System.out.println(" | 3 | Eliminar Encomendas |");
1457         System.out.println(" | 4 | Lista de Encomendas |");
1458         System.out.println(" | 0 | MENU |");
1459         System.out.println("+-----+");
1460         System.out.print("Escolha uma das seguintes opções: ");
1461         menuCliente = sc.nextInt();
1462         System.out.println("\n");
1463         switch (menuCliente) {
1464             case 1:
1465                 System.out.println("----- REGISTAR ENCOMENDAS ----- ");
1466                 try {
1467                     Encomenda temp = new Encomenda();
1468                     Produto produto = new Produto();
1469                     temp.id_encomenda = nEncomenda++;
1470                     System.out.print("Insira o NIF do cliente: ");
1471                     temp.nif_cliente = sc.nextInt();
1472                     sc.nextLine();
1473                     adicionarEncomenda(listaencomendas, temp);
1474                     int pos = pesquisarCliente(listaclientes, temp.nif_cliente);
1475                     if (pos != -1) {

```

```

1476
1477         int opcao = 1;
1478         while (opcao == 1) {
1479             System.out.print("Insira o ID Produto: ");
1480             produto.codigo = sc.nextInt();
1481             System.out.print("Inserir a quantidade pretendida: ");
1482             produto.quantidade = sc.nextInt();
1483             adicionarProdutosEncomenda(listaencomendas, pos, produto);
1484             System.out.println("Encomenda adicionada com sucesso!\n");
1485             System.out.print("Pretende inserir mais produtos? (1 - sim, 0 - não): ");
1486             opcao = sc.nextInt();
1487             sc.nextLine();
1488         }
1489     } else {
1490         System.out.println("O NIF introduzido não está registado no sistema");
1491     }
1492 } catch (Exception e) {
1493 }
1494 criarFicheiroEncomenda(listaencomendas);
1495 break;
case 2:
    System.out.println("----- EDITAR ENCOMENDAS -----");
    editarEncomendas(sc, listaencomendas);
    break;
case 3:
    System.out.println("----- ELIMINAR ENCOMENDAS -----");
    System.out.print("Inserir ID da Encomenda: ");
    int idRemover = sc.nextInt();
    removerEncomendas(listaencomendas, idRemover);
    break;
case 4:
    System.out.println("----- LISTA ENCOMENDAS -----");
    imprimirEncomendas(listaencomendas);
    break;
case 0:
    System.out.println("MENU");
    break;
default:
    System.out.println("Opção inválida!");
}
}while(menuCliente!=0);
break;
case 0:
    System.out.println("Até á proxima!!! ");
    break;
default:
    System.out.println("Opção inválida!");
}
} while (menu!=0);
}
}

```