

**Universidade do Minho**

*Escola de Engenharia*

Departamento de Informática

Licenciatura em Engenharia Informática

## **Projeto de LI3**

**“Transitórios LEI”**

***André Santos A60994***

***Helena Alves A61000***

***Ricardo Branco A61075***

Braga, março de 2012

## Resumo

Neste relatório está descrito todo o desenvolvimento do projecto “Transitórios LEI” desta unidade curricular. Começando pelas nossas escolhas, a escolha de uma ou várias estruturas foi um dos aspectos mais importantes na realização deste projecto.

Como vai ser explorado mais adiante neste relatório, houve em atenção a escolha da melhor estrutura de forma em que a execução do código fosse a mais rápida possível, nas operações mais importantes do programa. Por exemplo é muito importante, que haja a preocupação de que o tempo de execução de uma futura função que faça um envio de uma encomenda entre duas localidades seja a menor possível, em vez de nos preocuparmos na complexidade de código que gera as estatísticas.

Outro aspecto, ao qual se teve atenção foi a repetição de código. Não faz sentido haver uma função que insira um cliente numa estrutura e outra função que insira um camião numa estrutura similar. Sendo assim optou-se por implementar um conjunto de módulos de estruturas genéricas.

Por último, ofereceu-se a funcionalidade de salvar e recuperar o estado da aplicação. Desta forma, evita-se que todas as operações realizadas durante a execução do programa sejam perdidas após o término deste.

# Índice

|                     |          |
|---------------------|----------|
| <b>Resumo .....</b> | <b>2</b> |
|---------------------|----------|

|                           |          |
|---------------------------|----------|
| <b>1 Introdução .....</b> | <b>5</b> |
|---------------------------|----------|

|                                   |          |
|-----------------------------------|----------|
| <b>2 Estruturas de dados.....</b> | <b>6</b> |
|-----------------------------------|----------|

|                                  |   |
|----------------------------------|---|
| 2.1...Estruturas genéricas ..... | 6 |
|----------------------------------|---|

|                          |   |
|--------------------------|---|
| 2.1.1 Lista Ligada ..... | 6 |
|--------------------------|---|

|                            |   |
|----------------------------|---|
| 2.1.2 Tabela de Hash ..... | 6 |
|----------------------------|---|

|                                    |   |
|------------------------------------|---|
| 2.2...Estruturas Específicas ..... | 7 |
|------------------------------------|---|

|                         |   |
|-------------------------|---|
| 2.2.1 Localidades ..... | 7 |
|-------------------------|---|

|                     |   |
|---------------------|---|
| 2.2.2 Camiões ..... | 9 |
|---------------------|---|

|                      |   |
|----------------------|---|
| 2.2.3 Clientes ..... | 9 |
|----------------------|---|

|                                |    |
|--------------------------------|----|
| 2.2.4 Dados Estatísticos ..... | 10 |
|--------------------------------|----|

|                             |    |
|-----------------------------|----|
| 2.3...Funcionalidades ..... | 12 |
|-----------------------------|----|

|                       |    |
|-----------------------|----|
| 2.3.1 Inserções ..... | 12 |
|-----------------------|----|

|                      |    |
|----------------------|----|
| 2.3.2 Remoções ..... | 12 |
|----------------------|----|

|                     |    |
|---------------------|----|
| 2.3.3 Edições ..... | 12 |
|---------------------|----|

|                      |    |
|----------------------|----|
| 2.3.4 Procuras ..... | 12 |
|----------------------|----|

|                              |    |
|------------------------------|----|
| 2.3.5 Efectuar pedidos ..... | 13 |
|------------------------------|----|

|                       |    |
|-----------------------|----|
| 2.3.6 Ficheiros ..... | 13 |
|-----------------------|----|

|                          |           |
|--------------------------|-----------|
| <b>3 Interface .....</b> | <b>14</b> |
|--------------------------|-----------|

|   |           |
|---|-----------|
| <b>4 Complexidade e desempenho.....</b> | <b>20</b> |
|---|-----------|

|                               |    |
|-------------------------------|----|
| 4.1 ..Tempos de execução..... | 20 |
|-------------------------------|----|

|                       |    |
|-----------------------|----|
| 4.1.1 Inserções ..... | 20 |
|-----------------------|----|

|                      |    |
|----------------------|----|
| 4.1.2 Remoções ..... | 20 |
|----------------------|----|

|                      |    |
|----------------------|----|
| 4.1.3 Procuras ..... | 20 |
|----------------------|----|

|                            |    |
|----------------------------|----|
| 4.1.4 Efetuar pedido ..... | 21 |
|----------------------------|----|

|                       |    |
|-----------------------|----|
| 4.1.5 Ficheiros ..... | 21 |
|-----------------------|----|

|                                    |    |
|------------------------------------|----|
| 4.2 ..Utilização de Recursos ..... | 21 |
|------------------------------------|----|

|                          |           |
|--------------------------|-----------|
| <b>5 Conclusão .....</b> | <b>23</b> |
|--------------------------|-----------|

|                                   |           |
|-----------------------------------|-----------|
| <b>6 Informação do grupo.....</b> | <b>24</b> |
|-----------------------------------|-----------|

|                  |    |
|------------------|----|
| 6.1 ..Fotos..... | 24 |
|------------------|----|

|                        |    |
|------------------------|----|
| 6.2 ..Informação ..... | 25 |
|------------------------|----|

## Índice Ilustrativo

|  |    |
|--|----|
| Ilustração 1 – Lista Ligada Genérica .....                               | 6  |
| Ilustração 2 – Tabela de Hash de Localidades .....                       | 7  |
| Ilustração 3 – Elementos estrutura Localidade. ....                      | 8  |
| Ilustração 4 – Tabela de Hash de Clientes. ....                          | 9  |
| Ilustração 5 – Tabela de Hash de Camiões. ....                           | 9  |
| Ilustração 6 – Elementos estrutura Camiões. ....                         | 10 |
| Ilustração 7 – Listas ligadas de clientes para dados estatísticos. ....  | 11 |
| Ilustração 8 – Lista Ligada de localidades para dados estatísticos. .... | 11 |
| Ilustração 9 – Menu principal. ....                                      | 14 |
| Ilustração 10 – Menu Procurar peido. ....                                | 14 |
| Ilustração 11 – Menu gerir pedidos. ....                                 | 14 |
| Ilustração 12 – Menu Inserir pedido .....                                | 14 |
| Ilustração 13 – Menu inserir localidade .....                            | 15 |
| Ilustração 14 – Menu Remover localidade .....                            | 15 |
| Ilustração 15 – Menu Inserir ligações entre localidades .....            | 15 |
| Ilustração 16 – Menu gerir localidades .....                             | 15 |
| Ilustração 17 – Menu inserir camiões .....                               | 16 |
| Ilustração 18 – Menu editar camiões .....                                | 16 |
| Ilustração 19 – Menu remover camiões .....                               | 16 |
| Ilustração 20 – Menu gerir camiões .....                                 | 16 |
| Ilustração 21 – Menu Remover ligações entre localidades .....            | 16 |
| Ilustração 22 – Menu procurar clientes .....                             | 17 |
| Ilustração 23 – Menu remover clientes .....                              | 17 |

|  |    |
|--|----|
| Ilustração 24 – Menu inserir clientes .....                  | 17 |
| Ilustração 25 – Menu gerir clientes .....                    | 17 |
| Ilustração 26 – Menu procurar camião .....                   | 17 |
| Ilustração 27 – Resultado da procura .....                   | 18 |
| Ilustração 28 - Menu procurar cliente por nome .....         | 18 |
| Ilustração 29 – Menu procurar cliente por contribuinte ..... | 18 |
| Ilustração 30 – Menu dados estatísticos .....                | 18 |
| Ilustração 31 – Menu estado actual .....                     | 18 |
| Ilustração 32 – Menu estatística clientes .....              | 19 |
| Ilustração 33 – Menu estatística localidades .....           | 19 |

# 1 Introdução

No âmbito da cadeira Laboratórios de Informática III, perante o problema apresentado iremos desenvolver em C um programa que vista ajudar uma empresa de transportes – Transitários LEI – com uma aplicação que permita registar a informação necessária para o seu negócio.

O objetivo é criar um programa que gere as localidades que trabalham com esta empresa e as suas ligações, os camiões e os clientes existentes nesta empresa. Permitindo, assim, que o utilizador possa interagir de forma fácil com as funcionalidades que a empresa disponibiliza, ou seja, com as funcionalidades que o programa disponibiliza.

Na primeira etapa, desenvolvemos as diferentes estruturas de dados para clientes, camiões, localidades e dados estatísticos, os comandos base do programa e as respetivas operações de manutenção. Nesta segunda etapa, houve alterações de estruturas face aos ficheiros de teste apresentados pelo professor, novas implementações de funções de forma a permitir a inserção e pesquisa de pedidos, e funções de salvaguarda e recuperação do estado do programa.

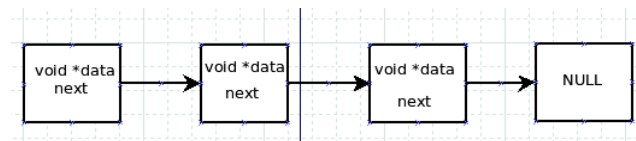
## 2 Estruturas de dados

Após a leitura e discussão do enunciado do projeto, decidimos utilizar dois tipos de estruturas: tabelas de Hash e listas ligadas. Estes dois tipos de estruturas foram criados de forma genérica de forma a tornar o projeto mais eficiente.

### 2.1 Estruturas genéricas

#### 2.1.1 Lista Ligada

Como a estrutura escolhida para armazenar os dados estatísticos foi lista ligada e, como também, iremos utilizar listas ligadas nas tabelas de Hash e em respetivos campos das estruturas específicas do projeto, então, decidimos criar uma lista genérica que iremos utilizar sempre que precisarmos trabalhar nesta estrutura.



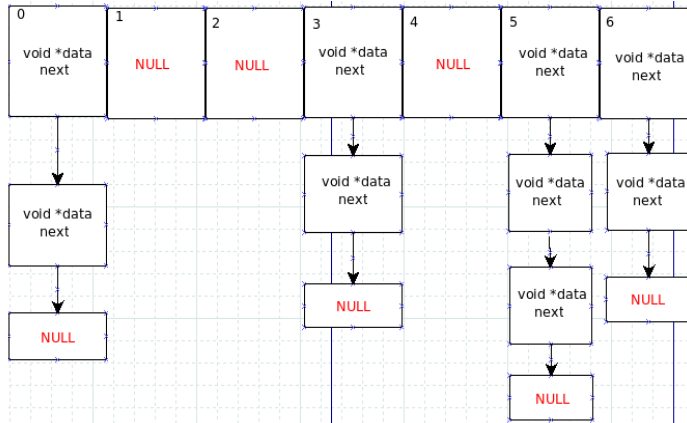
**Ilustração 1** – Lista Ligada Genérica

Assim, como está ilustrado na figura, podemos verificar que temos o tipo de dados (void\*) que pode ser utilizado como qualquer tipo de dados.

#### 2.1.2 Tabela de Hash

Como a estrutura escolhida para armazenar as localidades, camiões e clientes foi tabela de Hash com opção de chaining para tratamento de colisões, decidimos também criar uma tabela de Hash genérica.

Nesta estrutura temos, entre os mais importantes, o campo nrelem que nos indica o número de elementos inseridos na tabela de Hash, nrtotal que nos indica o número de posições do array, o apontador para a nossa função de Hash e o apontador para a função de comparação.



Escolhemos esta estrutura de dados pois pretendemos ter um tempo de acesso aos dados bastante reduzido, sendo que o tempo desejado de acesso é um tempo constante. Assim sendo, como escolhemos opção chaining, no melhor caso, o nosso tempo de pesquisa é constante e, no pior caso, é o número de nodos numa dada posição do array. Esperamos assim, ter um bom desempenho em termos de acesso à informação armazenada. Deste modo, temos também definido, que quando a nossa tabela de Hash se encontra ao nível de 80% de ocupação o número de posições do array duplica.

## 2.2 Estruturas Específicas

### 2.2.1 Localidades

Em relação às localidades, estas estão guardadas numa tabela de Hash com a adaptação da nossa tabela de Hash genérica criada.

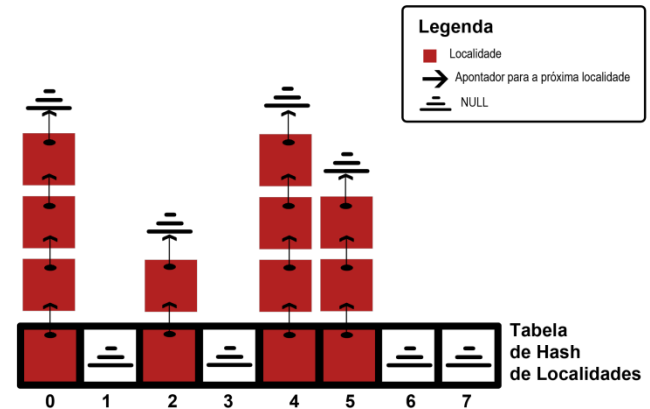


Ilustração 2 – Tabela de Hash de Localidades

Para cada localidade, existe uma estrutura *Localidade* que contém as informações desta: identificador da localidade, números de serviços na localidade, um apontador para uma lista ligada de clientes dessa localidade, um apontador para uma lista de camiões que se encontram nessa localidade, um apontador para uma lista ligada de localidades adjacentes onde a localidade em causa seja a localidade de partida e, por fim, um apontador para uma lista ligada de localidades adjacentes onde a localidade em causa seja a

localidade de destino.

Nesta segunda etapa, o nome da localidade foi substituído pelo seu identificador para permitir que existam localidades com o mesmo nome e, assim sendo, assumimos que o identificador de uma localidade seja o código postal da mesma.

As ligações entre localidades são armazenadas numa lista ligada, *Ligacoes*, que contém o nome da localidade de destino da ligação, a distância entre as duas localidades e as taxas associadas à ligação.

As localidades adjacentes a uma certa localidade são armazenadas numa lista ligada, *Adjacente*, que contém o nome da localidade adjacente.

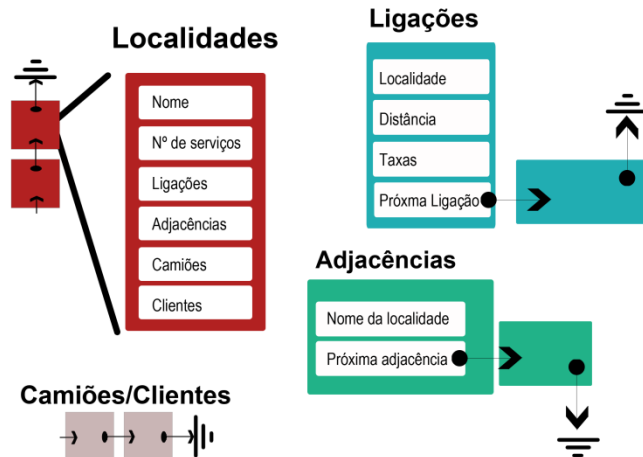


Ilustração 3 – Elementos estrutura Localidade.



## 2.2.2 Camiões

Falando agora dos camiões, estes também estão guardados numa tabela de Hash, adaptando novamente a nossa estrutura de Hash genérica.

Para cada Camião, existe uma estrutura *Camiao* que contém as informações deste: a identificação do camião, neste caso, o modelo, ano e cor do camião, a matrícula, os gastos por quilómetro, a capacidade total e o identificador da localidade onde se encontra o camião em determinado momento.

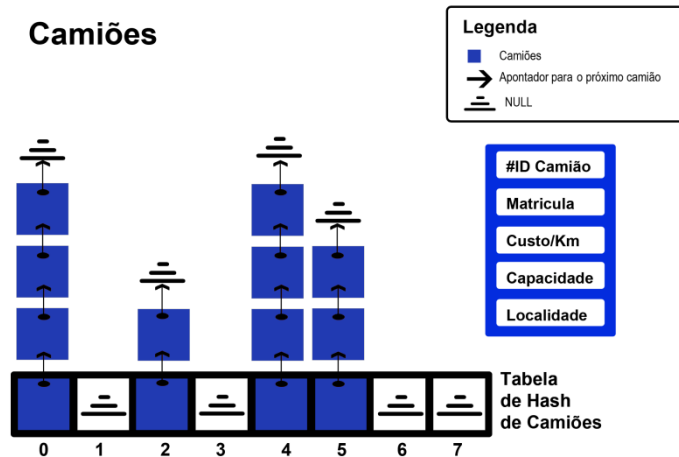


Ilustração 5 – Tabela de Hash de Camiões.

## 2.2.3 Clientes

Relativamente aos clientes, estes também estão guardados numa tabela de Hash, adaptando novamente a nossa estrutura de Hash genérica.

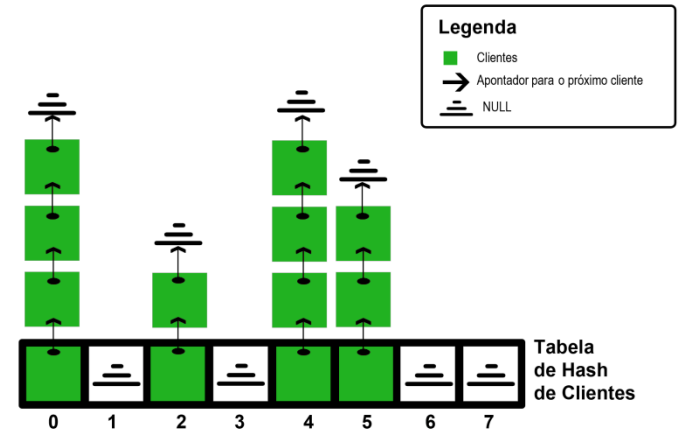
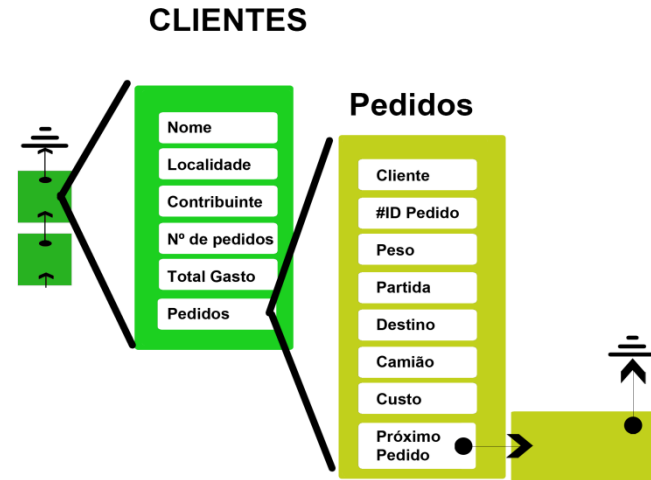


Ilustração 4 – Tabela de Hash de Clientes.

*Cliente* é a estrutura que contém as informações de cada cliente, tais como: nome, morada e contribuinte do cliente, número de pedidos do cliente, quantia total que já gastou em encomendas e um apontador para uma lista ligada de pedidos.



Os pedidos de cada cliente são armazenados numa lista ligada, *Pedidos*, que contém o nome do cliente que efetuou o pedido, a identificação do pedido, o seu peso, o identificador da localidade de partida e de destino, o custo do pedido, o camião que irá efetuar o pedido e os quilómetros que o camião vai percorrer desde a localidade de partida até à localidade de destino.

#### 2.2.4 Dados Estatísticos

Os nossos dados estatísticos estão armazenados em listas ligadas. Decidimos criar duas listas ligadas para as estatísticas dos clientes, uma lista ligada que guarda clientes por ordem decrescente dos seus gastos em encomendas, *EClienteg*, e outra que guarda os clientes por ordem decrescente da quantidade de número de pedidos efetuados, *EClientep*.

**Ilustração 6** – Elementos estrutura Camiões.

## ESTATÍSTICAS

### Clientes

#### -Pedidos



#### -Gastos



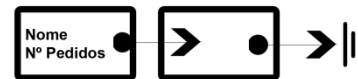
**Ilustração 7** – Listas ligadas de clientes para dados estatísticos.

Decidimos, também, criar duas listas ligadas para as estatísticas das localidades, uma lista que guarda as localidades por ordem crescente dos números de pedidos efetuados nessa localidade, e outra lista que guarda as localidades por ordem decrescente, também do número de pedidos efetuados.

## ESTATÍSTICAS

### Localidades

#### -Pedidos



**Ilustração 8** – Lista Ligada de localidades para dados estatísticos.

As estatísticas do nosso projeto só são geradas quando é acedido o menu “Gerir estatísticas”, isto porque as inserções ordenadas em listas ligadas são muito ineficientes e, como o menu “Gerir estatísticas” é muito menos acedido que o de inserções e remoções, concluímos que esta era a melhor solução.

## 2.3 Funcionalidades

### 2.3.1 Inserções

Neste programa, as inserções estão divididas em três módulos. O primeiro módulo interage com o utilizador, o outro lê os dados que o utilizador quer inserir e, por fim, o último utiliza as funções dos módulos genéricos para fazer a inserção dos dados inseridos.

Quando é realizada uma inserção são feitas cópias dos dados que foram inseridos para a nossa estrutura. É de salientar que são feitas cópias dos dados em vez de partilha com o sentido de preservar o encapsulamento do programa.

É importante realçar que tentamos praticar uma programação defensiva, e um dos exemplos são os testes de alocação de memória.

### 2.3.2 Remoções

Relativamente às remoções, estas também estão divididas em três módulos de forma similar às inserções.

Quando é realizada uma remoção, o elemento a ser removido é eliminado em todas as estruturas em que esteja inserido. Com por exemplo, quando é removido

um cliente, este é removido da respectiva tabela de hash e, também, das listas ligadas dos dados estatísticos.

### 2.3.3 Edições

Relativamente às edições, estas também estão divididas em três módulos como as inserções e remoções.

Quando uma edição é realizada, o programa lê os novos dados a ser inseridos e insere-os na respectiva estrutura.

### 2.3.4 Procuras

Neste programa, todas as categorias do programa oferecem a opção de procura, desde a procura de um camião, procura de um cliente e até mesmo procura de um pedido. Todos os pedidos usam as funções de procura oferecidos pelos módulos genéricos.

Pode-se distinguir 2 tipos de procuras: as procuras de clientes, camiões e localidades e a procura de pedidos, estes apoiam-se sobre a mesma estrutura de dados, mas oferecem tempos de pesquisa bastante diferente. Por exemplo, no caso dos clientes, a procura é feita em função da função de hash, sendo que a procura acaba por ser quase constante, o mesmo já não acontece com a pesquisa de pedidos, uma vez que o critério de procura é um substring da descrição do mesmo, o que

implica que a tabela de Hash tenha que de ser toda percorrida e que por sua vez o tempo de procura fique longe de ser constante, dependendo assim no numero de elementos da tabela.

### 2.3.5 *Efectuar pedidos*

De todas as funcionalidades oferecidas pelo programa, a realização de um pedido caracteriza-se por ter um comportamento "especial" durante a execução do programa.

De forma a minimizar o tempo de execução do mesmo, houve uma maior preocupação na escolha de um algoritmo e adoção do mesmo. O algoritmo escolhido foi baseado no algoritmo de Dijkstra, um algoritmo de dados que dado um grafo conecto e com pesos não negativos, calcula o caminho mais curto de um nodo para todos os nodos.

Houve no entanto alteração do algoritmo para que este estivesse de "acordo" com a estrutura do programa e, por consequência, este ofereceu um tempo de execução inferior ao algoritmo original, uma vez que o algoritmo pára assim que o nodo de destino é atingido.

É importante realçar que na fase inicial do pedido, na procura do camião para realizar o serviço, foi feita uma versão do mesmo algoritmo em que este pára

mal encontre um camião numa localidade já visitada. Esta implementação vai permitir reduzir significativamente o tempo a efetuar um pedido visto que, assim, não precisamos de calcular o caminho mais curto de todos os camiões para a localidade de partida.

Se associarmos o problema à realidade, a nossa implementação faz todo sentido, visto que é mais vantajoso usar um camião mais próximo da localidade do que transportar um camião de uma localidade distante, pois aí também estaríamos a perder mais tempo.

### 2.3.6 *Ficheiros*

Em relação aos ficheiros, o programa adopta a serialização de dados. O programa escreve as estruturas em modo binário para assegurar maior rapidez e maior segurança pois os dados são mais difíceis de editar por meios exteriores ao programa.

Após a leitura das estruturas base, como estas incluem outras estruturas de apontadores, estes são actualizados através de ciclos.

Sempre que iniciamos o programa, este carrega os dados dos ficheiros para as estruturas, e quando o encerramos, este sistema grava as estruturas em ficheiros.

### 3 Interface

```
##### TRANSITÁRIOS LEI #####
#
# 1 - Gerir pedidos
# 2 - Gerir localidades
# 3 - Gerir camiões
# 4 - Gerir clientes
# 5 - Dados Estatísticos
# 6 - Sair
#
#####
```

Ilustração 9 – Menu principal.

```
##### TRANSITÁRIOS LEI #####
#
# 1 - Efectuar pedido
# 2 - Procurar pedido
# 3 - Voltar atrás
# 4 - Sair
#
##### Procurar Clientes #####
```

Ilustração 11 – Menu gerir pedidos

```
##### INSERIR PEDIDO #####
* Insira um novo pedido no seguinte formato:
* Contribuinte | ID Partida | ID Destino | Peso | Descrição.
* Linha em branco para terminar.
```

Ilustração 12 – Menu Inserir pedido

```
##### PEDIDO EFECTUADO #####
Camiao utilizado: 22-54-GH
Custo por km: 7.100000
Localidade anterior: 12689
Distancia para a origem da entrega: 699.00
Custo minimo até a origem da entrega: 5064.90

Distancia entre a origem e o destino da entrega: 1047.00
Custo minimo entre a origem e o destino da entrega: 7546.70

Distancia total: 1746.00
Custo total: 12611.60
```

```
##### PROCURAR PEDIDO #####
* Insira a descrição do pedido.
* Linha em branco para terminar.
```

Ilustração 10 – Menu Procurar pedido.

```
##### PROCURAR PEDIDOS #####

Nº de contribuinte: 9000011000
Descrição do Pedido: Materiais
Partida: 200
Destino: 300
Peso da Mercadoria: 23.00
Matriculo do camião: 22-54-GH
Nº de Km percorridos 1746.00
Custo da mercadoria 12611.60

Nº de contribuinte: 9000021664
Descrição do Pedido: Materiais de Construção
Partida: 16957
Destino: 1886
Peso da Mercadoria: 11.00
Matriculo do camião: 22-87-VG
Nº de Km percorridos 1728.00
Custo da mercadoria 13096.00
```

Quando é pedido ao utilizador para procurar um pedido, este ao inserir o nome do pedido são dadas como resposta todos os pedidos que contêm esse nome, como se vê no exemplo.

```
##### TRANSITÁRIOS LEI #####
#                                     #
# 1 - Inserir localidade             #
# 2 - Remover localidade             #
# 3 - Gerir ligações entre localidades #
# 4 - Procurar localidades           #
# 5 - Voltar atrás                   #
# 6 - Sair                           #
#                                     #
##### Gerir Localidades #####
```

**Ilustração 16** – Menu gerir localidades

```
##### INSERIR LOCALIDADES #####
* Insira uma localidade por linha no seguinte formato.
* Nome Localidade | Identificador.
* Linha em branco para terminar.

Lisboa | 200
E_EXIST: Elemento já existe
Guimarães | 19000
E_SUCESS: Operação realizada com êxito
_
```

**Ilustração 13** – Menu inserir localidade

Neste menu, optamos por permitir ao utilizador inserir uma localidade por linha e linha branca para voltar ao menu. São apresentadas mensagens de sucesso no ecrã quando é possível inserir localidade e, quando não é possível é apresentada mensagem com o respetivo erro.

```
##### REMOVER LOCALIDADES #####
* Remova uma localidade por linha, escreva o identificador da localidade.
* Linha em branco para terminar.

19000
E_SUCESS: Operação realizada com êxito
20000
E_PROC: Elemento nao existe
_
```

**Ilustração 14** – Menu Remover localidade

```
##### INSERIR LIGAÇÕES ENTRE LOCALIDADES #####
* Insira o identificador da localidade que quer adicionar ligações.
* Linha em branco para terminar.

200_
```

**Ilustração 15** – Menu Inserir ligações entre localidades

Neste menu - Inserir Ligações entre localidades -, inicialmente insere-se a localidade onde se quer adicionar ligação e, posteriormente aparece outro menu para o utilizador digitar a localidade de destino da ligação e características da ligação.

```
##### REMOVER LIGAÇÕES DA LOCALIDADE #####
* Insira o id das ligações da localidade para remover.
* Linha em branco para terminar.

250
E_SUCESS: Operação realizada com êxito
```

```
##### REMOVER LIGAÇÕES DA LOCALIDADE #####
* Insira o id das ligações da localidade para remover.
* Linha em branco para terminar.

250
E_SUCESS: Operação realizada com êxito
340
E_PROC: Elemento nao existe
```

**Ilustração 21** – Menu Remover ligações entre localidades

```
##### TRANSITÁRIOS LEI #####
#                                     #
# 1 - Inserir camião                 #
# 2 - Remover camião                 #
# 3 - Alterar informação de um camião #
# 4 - Procurar camião                #
# 5 - Voltar atrás                   #
# 6 - Sair                           #
#                                     #
##### Gerir Camiões #####
```

**Ilustração 20** – Menu gerir camiões

```
##### INSERIR CAMIOES #####
* Insira um camião com o seguinte formato:
* Id Localidade | Matricula | Custo km | Capacidade | Informacoes
* Capacidade - toneladas, Custo - euros.
* Formato exemplo: 100 | 00-00-AA | 10 | 3.4 | Mercedes,Azul,2012
* Linha em branco para terminar.

200 | 22-22-AS | 2.2 | 70 | Azul,Mercedes,2003
E_SUCESS: Operação realizada com êxito
300 | 22-22-AS | 3.4 | 120 | Amarelo,Toyota,1999
E_EXIST: Elemento já existe
```

**Ilustração 17** – Menu inserir camiões

```
##### REMOVER CAMIOES #####
* Insira a matrícula do camião a remover.
* Linha em branco para terminar.

22-22-AS
E_SUCESS: Operação realizada com êxito
44-99-TG
E_PROC: Elemento nao existe
```

**Ilustração 19** – Menu remover camiões

```
##### EDITAR CAMIOES #####
* Insira os novos dados dos camiões com o seguinte formato:
* Identificador da localidade | Matricula | Custo km | Capacidade | Informações
* Capacidade - toneladas, Custo - euros.
* Formato exemplo: Lisboa | 00-00-AA | 10 | 3,4 | Mercedes,Azul,2012
* Linha em branco para terminar.

200 | 22-22-AS | 3.6 | 80 | Azul,Mercedes,2006
E_PROC: Elemento nao existe
```

**Ilustração 18** – Menu editar camiões



```
##### PROCURAR CAMIÕES #####
* Insira a matrícula do camião.
* Linha em branco para terminar.
```

```
##### PROCURAR CAMIÕES #####

Identificador: Azul,Opel,2003 ;
Matrícula: 78-12-WE ;
Custo por quilómetro: 5.50 ;
Capacidade total: 150.00 ;
Localidade onde se encontra: 350 ;
```

Ilustração 26 – Menu procurar camião

```
##### TRANSITÁRIOS LEI #####
#                                     #
# 1 - Inserir cliente                 #
# 2 - Remover cliente                 #
# 3 - Alterar informação de um cliente #
# 4 - Procurar Cliente                #
# 5 - Voltar atrás                    #
# 6 - Sair                            #
#                                     #
##### Gerir Clientes #####
```

Ilustração 25 – Menu gerir clientes

```
##### INSERIR CLIENTES #####
* Insira um cliente com o seguinte formato:
* Nome | Mail | Morada do cliente | Contribuinte
* Formato exemplo: António | antonio22@gmail.com | Torres Novas | 131456263
* Linha em branco para terminar.

Antonio | ads@gmail.com | Guimarães | 1233456
E_SUCESS: Operação realizada com êxito
Susana | edr@gmail.com | Porto | 1233456
E_EXIST: Elemento já existe
```

Ilustração 24 – Menu inserir clientes

```
##### REMOVER CLIENTES #####
* Insira o contribuinte do cliente que quer remover.
* Linha em branco para terminar.

1233456
E_SUCESS: Operação realizada com êxito
5678783
E_PROC: Elemento nao existe
```

Ilustração 23 – Menu remover clientes

```
##### TRANSITÁRIOS LEI #####
#                                     #
# 1 - Procurar por contribuinte      #
# 2 - Procurar por nome               #
# 3 - Voltar atrás                    #
# 4 - Sair                            #
#                                     #
##### Procurar Clientes #####
```

Ilustração 22 – Menu procurar clientes

```
##### PROCURAR CLIENTES #####
* Insira o contribuinte do cliente.
* Linha em branco para terminar.

9000011000
```

Ilustração 29 – Menu procurar cliente por contribuinte

```
##### PROCURAR CLIENTES #####
* Insira o nome do cliente.
* Linha em branco para terminar.

Jon Yang
```

Ilustração 28 – Menu procurar cliente por nome

```
##### PROCURAR CLIENTES #####

Nome do Cliente: Jon Yang
Localidade: 3761 N. 14th St
Mail: 9000011000@foursquareUM.com
Contribuinte: 9000011000
Número pedidos e total gasto: 1 12611.60
##### ULTIMO PEDIDO #####
Partida: 200
Destino: 300
Peso da Ecomenda: 23.00
Custo da Ecomenda: 12611.60
Camião: 22-54-GH
Descrição: Materiais
```

Ilustração 27 – Resultado da procura

```
##### TRANSITÁRIOS LEI #####
#
# 1 - Estado actual
# 2 - Estatística clientes
# 3 - Estatística localidades
# 4 - Voltar atrás
# 5 - Sair
#
##### Gerir Dados Estatísticos #####
```

Ilustração 30 – Menu dados estatísticos

```
##### ESTADO GERAL #####
##### LOCALIDADES #####

Nº de localidades: 18515
Nº de ligações: 166580
Localidade mais requisitada: 62      2
Localidade menos requisitada: 17999  0

##### CLIENTES #####

Nº de clientes: 18484
Clientes com mais ecomendas 9000028280 3
Cliente com mais gastos: 9000028280 23836.10

##### CAMIÕES #####

Nº de Camiões: 40

##### PEDIDOS #####

Pedido mais caro: 9000028280 23836.10
Pedido mais longo: 9000016800 2821.00
Pedido mais pesado: 9000028280 77.00
```

Ilustração 31 – Menu estado actual

```
##### CLIENTES #####
##### CLIENTES COM MAIS ENCOMENDAS #####

** 9000028280    3
** 9000023872    1
** 9000023471    1
** 9000023427    1
** 9000023399    1
** 9000023000    1
** 9000022261    1
** 9000021664    1
** 9000021264    1
** 9000020962    1

##### CLIENTES COM MAIS GASTOS #####

** 9000028280    23836.100
** 9000020534    17269.299
** 9000017816    17081.000
** 9000026944    16980.000
** 9000027152    15735.000
** 9000028292    14871.000
** 9000021664    13096.000
** 9000011000    12611.600
** 9000027231    12486.600
** 9000021264    12386.400
```

Ilustração 32 – Menu estatística clientes

```
##### LOCALIDADES #####
##### LOCALIDADES MAIS REQUISITADAS #####

**          62      2
**          41      2
**         16957    1
**         17817    1
**         17474    1
**         15946    1
**         17456    1
**         16904    1
**         14569    1
**         13844    1

##### LOCALIDADES MENOS REQUISITADAS #####

**          17999    0
**          17998    0
**          17989    0
**          17997    0
**          17988    0
**          17996    0
**          17899    0
**          17979    0
**          17987    0
**          17995    0
```

Ilustração 33 – Menu estatística localidades

## 4 Complexidade e desempenho

### 4.1 Tempos de execução

#### 4.1.1 Inserções

As inserções deste programa apresentam tempos iguais às inserções numa tabela de hash, ou seja, apresentam tempos quase constantes e quase imediatos.

Apesar de nas inserções de clientes ser necessário inserir o cliente em duas tabelas de hash, como também nas inserções de camiões ser necessário inserir o camião na tabela de hash e na respectiva localidade, o tempo de execução não se altera significativamente, continuando a ser quase imediato.

Nas inserções de ligações, à semelhança de outras inserções são feitas de forma ordenada em listas ligadas e, apesar de prejudicar o desempenho da inserção, não vão alterar significativamente o tempo de execução, devido a estas listas serem normalmente curtas e, ainda, vão melhorar outras funcionalidades.

Por último, as inserções de pedidos são feitas à cabeça de uma lista ligada, vão apresetar tempo constante.

As inserções no nosso programa são quase todas imediatas, que significa que fornece ao utilizador uma velocidade de inserção excelente.

#### 4.1.2 Remoções

As remoções deste programa apresentam tempos iguais às inserções. O tempo de execução das diferentes remoções do programa são similares às inserções das mesmas. Por isso, as remoções apresentam também tempos quase imediatos.

#### 4.1.3 Procuras

As procuras deste programa apresentam também tempos quase constantes e quase imediatos.

Apenas existe uma diferença na procura de pedidos, sendo esta mais ineficiente pois percorre toda a tabela de hash de clientes e, por consequência, as suas listas de pedidos. Apesar disto, a procura apresenta tempos quase imediatos.

#### 4.1.4 Efetuar pedido

*Efetuar pedido* é a funcionalidade mais lenta do programa pois conjuga várias inserções e procuras.

O cálculo do caminho mais curto é demorado e, apesar das otimizações feitas ao algoritmo, este ainda apresenta um tempo bastante superior aos tempos das restantes funcionalidades. No entanto, a nossa estrutura e as nossas optimizações permitiram que o tempo fosse reduzido, obtendo assim um tempo médio bastante bom, por volta dos 5 segundos.

Quando se efectua um pedido, os tempos de inserções e procuras são insignificantes comparativamente ao tempo de procura do caminho mais curto.

Concluindo, o tempo médio de 5 segundos é bastante aceitável tendo em o que a funcionalidade oferece.

#### 4.1.5 Ficheiros

Ao ser implementado uma escrita e leitura de ficheiro em modo binário asseguramos também, nestas funcionalidades, um tempo de execução superior às funcionalidades anteriormente referidas, com a exceção do *Efectuar pedido*.

O tempo de leitura ronda os 0.3 segundos e o tempo de escrita ronda os 0.1 segundos. Estes são tempos excelentes principalmente para funcionalidades que são executadas poucas vezes no sistema.

## 4.2 Utilização de Recursos

Os valores indicados na subsecção 4.1 (Tempos de execução) foram obtidos num PC com um processador com 2 núcleos e frequência de 2.0 GHz e com 4 Gb de memória. Face a execução do programa os recursos utilizados foram os seguintes:

| RECURSO            | Tamanho (Mb) |
|--------------------|--------------|
| Memória Virtual    | 142          |
| Memória Física     | 134          |
| Memória Partilhada | 432          |
| Código             | 48           |
| Dados              | 138          |

Sendo assim e face aos valores apresentados e relacionando-os com as características do computador justifica-se os tempos de execução obtidos na subsecção 4.1 devido as características do processador e que os gastos de memória se justificam devido a quantidade de estruturas de dados utilizadas e à quantidade de informação inserida nas mesmas.

## 5 Conclusão

Neste projeto, ao escolhermos trabalhar com tabelas de Hash assegurou-se que o programa, mesmo com um elevado número de inserções de localidades, camiões e clientes o nosso programa consiga um bom desempenho em termos de pesquisa e acesso à informação.

No entanto, as listas ligadas dos dados estatísticos, com um grande número de elementos, pode fazer com que o programa fique um pouco mais lento, mas como estas só são acedidas no menu “Gerir estatísticas”, não terão muita interferência no funcionamento do programa, visto que estas precisam de ser acedidas poucas vezes.

A interface utilizada é bastante amigável para o utilizador, e permite uma utilização rápida e é adaptada as funcionalidades mais utilizadas.

O programa apresenta tempos quase imediatos em todas as pesquisas, inserções e remoções.

A adaptação do algoritmo Djisktra ao nosso programa/estrutura pensamos que foi uma boa escolha

pois veio assegurar rapidez e solidez nas inserções de pedidos.

Os gastos de memória são justificáveis face à estrutura interna do programa.

Em geral, com as estruturas e funcionalidades escolhidas, iremos conseguir um bom funcionamento do nosso programa.

## 6 Informação do grupo

### 6.1 Fotos



André Santos  
A60994



Helena Alves  
A61000



Ricardo Branco  
A61075



## 6.2 Informação

**André Santos:** andreccdr@gmail.com.

Escola EB 2,3 Ribeirão.

Escola Secundária D. Sancho I, Famalicão – Ciências e Tecnologias.

Interesse nas áreas de programação móvel, programação web, gaming, novas tecnologias, informática em geral.

**Helena Alves:** helenalves4@gmail.com.

Escola EB 2,3 S. Paio Moreira de Cónegos.

Escola Secundária Caldas de Vizela – Ciências e Tecnologias.

Interesse em atletismo e desporto em geral.

**Ricardo Branco:** ricax92@gmail.com.

Escola EB 2,3/S Pintor José de Brito, Viana do Castelo.

Escola Secundária de Monserrate, Viana do Castelo – Técnico Profissional de Gestão de Equipamentos Informáticos.

Interesse em redes, novas tecnologias e xadrez.