

**Universidade do Minho**  
**Licenciatura em Ciências da Computação**  
**2º ano**  
**2019/2020**

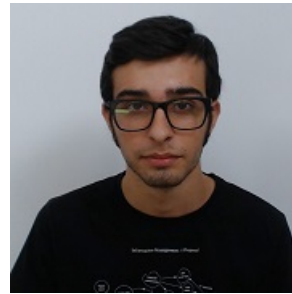
**Trabalho Prático**  
**Programação Orientada a Objetos**  
**Grupo 18**



André Silva (a87958)



João Silva (a87939)



João Gouveia (a87995)

# Conteúdo

<b>1. Introdução</b>	<b>4</b>
1.1. Estratégias para a criação da aplicação	4
<b>2. Classes</b>	<b>5</b>
2.1. Classes principais	5
2.1.1. Ponto	5
2.1.2. LinhaEncomenda	6
2.1.3. Encomenda	6
2.1.4. EncomendasAceites	6
2.1.5. Utilizador	7
2.1.6. PlataformaEntrega	7
a) Transportadoras	8
b) Voluntarios	8
2.1.7. Lojas	8
2.1.8. TrazAqui	9
2.2. Classes de exceções	9
2.2.1. exCodigoAlreadyExists	9
2.2.2. exCodigoDoesNotExist	9
2.2.3. exEmailAlreadyExists	10
2.2.4. exEmailDoesNotExist	10
2.2.5. exInvalidNIF	10
2.2.6. exLojaAlreadyExists	10
2.2.7. exNegativeValues	10
2.2.8. exPEAlreadyExists	10
2.2.9. exPENotAvailable	11
2.2.10. exPWIncorrect	11
2.3. Classes de comparadores	11
2.3.1. cMelhorClassificado	11
2.3.2. cMelhorCusto	11
2.3.3. cTopPE	11
2.3.4. cTopU	12
<b>3. Interação com a aplicação</b>	<b>12</b>
3.1. Menu Inicial	12
3.2. Sign up	13
3.2.1. Sign Up – Utilizador	13
3.2.2. Sign Up – Loja	13
3.2.3. Sign Up – Transportadora/Voluntário	14
3.3. Leaderboards	15
3.4. Log In	15
3.5. Utilizador	15
3.5.1. Utilizador – Menu principal	15
3.5.2. Utilizador – A minha encomenda	16
1) Ver estado da encomenda	16
2) Inserir pedido de encomenda	16
3) Pedir transporte	17
4) Classificar o serviço de transporte	18
3.5.3. Utilizador – Definições	19

3.5.4. Utilizador – Informações adicionais	19
3.6. Loja	20
3.6.1. Loja – Menu Principal	20
3.6.2. Loja – Número de pessoas atualmente na fila	20
3.6.3. Loja – Registo de encomendas entregues	20
3.7. Transportadora	21
3.7.1. Transportadora – Menu Principal	21
3.7.2. Transportadora – Total faturado	21
3.7.3. Transportadora – Classificação atual da empresa	21
3.7.4. Transportadora – Registo de entregas realizadas	22
3.7.5. Transportadora – Definições	22
3.8. Voluntário	22
3.8.1. Voluntário – Menu Principal	23
3.8.2. Voluntário – Classificação atual	23
3.8.3. Voluntário – Registo das entregas realizadas	23
3.8.4. Voluntário – Definições	23
<b>4. Conclusão</b>	<b>24</b>

# 1. Introdução

Neste relatório, é descrito o funcionamento da aplicação que desenvolvemos para a UC de Programação Orientada a Objetos, descrevendo as classes e as estratégias utilizadas pelo nosso grupo para resolver o problema proposto.

Como é explicado em pormenor em seguida, este projeto baseia-se na criação de uma aplicação de entrega de encomendas com várias funcionalidades envolvendo clientes que recebem encomendas, lojas, empresas e voluntários que transportam essas encomendas.

## 1.1. Estratégias para a criação da aplicação

Durante a criação desta aplicação, é possível dividir a estratégia que nós utilizamos em três partes distintas.

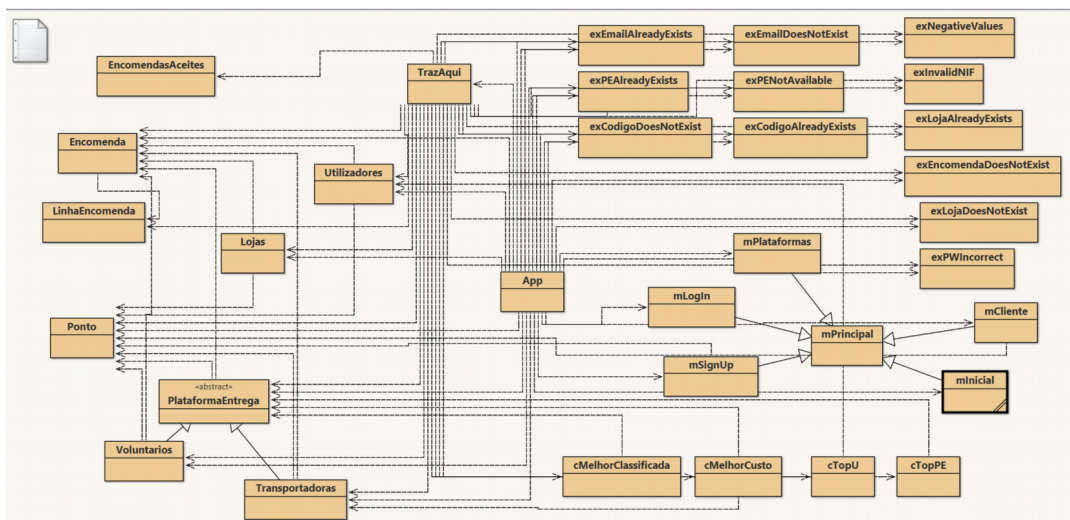
Numa fase inicial, nós criamos as classes principais e os métodos que consideramos importantes para o funcionamento da aplicação de modo a completar os requisitos propostos.

À medida que os métodos eram criados, fomos modificando as classes e procedemos à criação de novas classes de modo a permitir o funcionamento dos métodos.

No momento em que consideramos que já estavam presentes todos os métodos necessários, procedemos à criação das classes e métodos que permitiam a interação entre utilizador e a aplicação. Essa interação é feita a partir de texto imprimido no ecrã com a possibilidade de escrever dígitos para aceder a um certo menu ou a introdução texto para completar uma certa tarefa (se for esse o caso).

Numa abordagem final, alteramos certos métodos de forma a serem compatíveis com o sistema de menus criados e, quando consideramos que os métodos estavam implementados corretamente, procedemos à correção de eventuais *bugs* que existiam enquanto testamos o programa.

## 2. Classes



Aqui apresentamos as classes que achamos indicadas para a concessão desta aplicação. Achamos pertinente dividir as classes em 3 grupos distintos: classes principais; classes de exceções e classes de comparadores.

### 2.1. Classes Principais

**Nota:** Antes de introduzirmos as seguintes classes, gostávamos de salientar que sempre que é mencionado um código (ou um e-mail), essa *String* é única para cada uma dessas classes, ou seja, não existem dois utilizadores com o mesmo código/email, não existem duas plataformas de entrega com o mesmo código/email, etc. O mesmo não acontece com o nome de utilizador. Dois utilizadores podem ter o mesmo nome e continuam a ser utilizadores válidos.

#### 2.1.1. Ponto

A classe *Ponto* foi desenvolvida de modo a satisfazer a necessidade da existência coordenadas de localização nas classes *Utilizador*, *Loja* e na classe abstrata *PlataformaEntrega* (todas estas descritas mais à frente). Esta classe contém as seguintes variáveis de instância:

```
private float x;  
private float y;
```

onde as variáveis **x** e **y** referem-se à coordenada GPS da longitude e à coordenada GPS da latitude, respetivamente.

### 2.1.2. LinhaEncomenda

A classe *LinhaEncomenda* é utilizada para auxiliar a classe *Encomenda* (que irá ser descrita em seguida).

Esta classe contém as seguintes variáveis de instância:

```
private String codProduto;  
private String descricao;  
private float quantidade;  
private float valorUnitario;
```

onde a variável **codProduto** representa o código do produto nessa linha de encomenda; **descricao** contém a descrição desse produto; **quantidade** tem a informação referente à quantidade desse produto existente na linha de encomenda; e **valorUnitario** refere-se ao preço do produto por unidade.

### 2.1.3. Encomenda

A classe *Encomenda* contém a informação de uma encomenda.

As variáveis de instância desta classe são as seguintes:

```
private String codEncomenda;  
private String codUtilizador;  
private String codLoja;  
private float peso;  
private LocalDateTime data;  
private List<LinhaEncomenda> encomendas;
```

onde a variável **codEncomenda** contém o código da encomenda; **codUtilizador** contém o código do utilizador que irá pedir a entrega dessa encomenda (a classe *Utilizador* é descrita no próximo ponto); **codLoja** contém o código da loja onde essa encomenda será realizada (a classe *Loja* estará, também, descrita mais à frente); **peso** tem o peso da encomenda; **data** contém a sua data de criação; e **encomendas** contém uma lista das *LinhaEncomenda* com os produtos da encomenda em questão.

### 2.1.4. EncomendasAceites

A classe *EncomendasAceites* é uma classe com apenas uma variável de instância:

```
private String codEncomenda;
```

esta variável representa o código de uma encomenda que já foi entregue ao utilizador.

### 2.1.5. Utilizadores

O nosso grupo definiu a classe *Utilizadores* como a classe que contém todas as informações e métodos referentes ao cliente que pede o transporte das encomendas.

Esta classe contém as seguintes variáveis de instância:

```
private String codigo;  
private String nome;  
private String email;  
private String pw;  
private Ponto GPS;  
private Map<String, Encomenda> encomendasGuardadas;
```

onde a variável **codigo** representa o código do utilizador; **nome** tem o nome do utilizador; email guarda o seu e-mail; **pw** contém a sua palavra passe; **GPS** é um *Ponto* com a sua localização geográfica; e **encomendasGuardadas** é um *Map* onde se encontram as encomendas efetuadas pelo utilizador.

### 2.1.6. PlataformaEntrega

Durante o desenvolvimento da aplicação, achamos que a forma mais simples de implementar as classes *Transportadoras* e *Voluntarios* seria com a utilização da classe abstrata *PlataformaEntrega*. *Transportadoras* e *Voluntarios* têm a maioria das funcionalidades em comum, então, na prática, estas duas classes acabam por ter apenas pequenos pormenores e maioria dos seus métodos e variáveis de instância estão incluídos em *PlataformaEntrega*.

Portanto, as variáveis de instância utilizadas nesta classe abstrata são:

```
private String codigo;  
private String password;  
private String nome;  
private Ponto GPS;  
private double raio;  
private boolean transporteMedicamentos;  
private boolean disponivel;  
private List<Integer> classificacoes;  
private Map<String, Encomenda> encomendasGuardadas;
```

onde a variável **codigo** guarda o código da empresa de transporte/ voluntário; **password** contém a sua palavra passe; nome contém o seu nome; **GPS** guarda num *Ponto* as suas

coordenadas geográficas; **raio** contém o raio máximo a qual a transportadora/ voluntário pode viajar para fazer o transporte da encomenda a partir da sua localização guardada em GPS; **transporteMedicamentos** indica se a empresa de transporte/ voluntário pode fazer ou não transporte de medicamentos; **disponivel** indica se a empresa de transporte/ voluntário se encontram disponíveis para realizar uma entrega; **classificacoes** é uma List que inclui todas as classificações feitas por utilizadores à empresa de transporte/ voluntário de forma a ser mais fácil de calcular a sua média; e **encomendasGuardadas** que é um Map onde são guardadas as encomendas transportadas pela empresa de transporte/ voluntário.

## a) Transportadoras

Esta classe existe para representar as empresas transportadoras de encomendas. Além das variáveis existentes em *PlataformaEntrega*, esta classe também inclui:

```
private String NIF;  
private double precoPorKM;  
private int capacidade;
```

onde a variável **NIF** contém o número de identificação fiscal da empresa; **precoPorKM** contém o preço do transporte das encomendas por quilómetro; e **capacidade** refere-se a quantos trabalhadores estão disponíveis para fazer o transporte de encomendas.

## b) Voluntarios

A classe *Voluntarios* foi criada para representar os voluntários que transportam as encomendas e não tem nenhuma variável de instância que não esteja já representada em *PlataformaEntrega*.

### 2.1.7. Lojas

A classe *Lojas* foi criada para podermos representar corretamente as lojas que possuem as encomendas a ser entregues.

Esta classe inclui as seguintes variáveis de instância:

```
private String codigo;  
private String password;  
private String nome;  
private Ponto GPS;  
private int numeroEncomendas;  
private Map<String, Encomenda> registoEncomendas;
```



onde a variável **codigo** guarda o código da loja; **password** guarda a sua password; **nome** guarda o nome da loja; **GPS** guarda as coordenadas geográficas da localização da loja; **numeroEncomendas** contém o número de pessoas que se encontram na fila da loja; e **registoEncomendas** guarda num *Map* as encomendas levantadas nessa loja.

### 2.1.8. TrazAqui

A classe TrazAqui é a classe central da aplicação, onde se encontram métodos que envolvem as outras classes. É também onde se encontram os métodos de ler, parse e guardar ficheiros. As variáveis de instância que esta classe inclui são as seguintes:

```
private Map<String, PlataformaEntrega> informacaoPlataformaEntrega;  
private Map<String, Lojas> informacaoLojas;  
private Map<String, Utilizadores> informacaoUtilizadores;  
private Map<String, Encomenda> informacaoEncomenda;  
private List<EncomendasAceites> encomendasRealizadas;
```

onde **informacaoPlataformaEntrega** é um Map que inclui todas os voluntários e empresas de transporte registadas no sistemas; **informacaoLojas** é um Map que inclui todas as lojas registadas no sistemas; **informacaoUtilizadores** é um Map que contém todos os utilizadores registados no sistemas; **informacaoEncomenda** é um Map que contém todas as encomendas registadas no sistemas; e **encomendasRealizadas** é uma List composta pelos códigos das encomendas que já foram entregues aos utilizadores.

## 2.2. Classes de Exceções

Enquanto desenvolvíamos a aplicação, sentimos a necessidade de criarmos classes para lidar com eventuais exceções que pudessem aparecer.

### 2.2.1. exCodigoAlreadyExists

Esta exceção refere-se a, no momento de criação de uma conta, ser introduzido um código já associado a outra conta da mesma classe.

### 2.2.2. exCodigoDoesNotExist

Esta exceção refere-se a, no momento de log in, ser introduzido um código não associado a nenhuma conta.

### 2.2.3. exEmailAlreadyExists

Esta exceção refere-se a, no momento de criação de uma conta, ser introduzido um email já associado a outra conta da mesma classe.

### 2.2.4. exEmailDoesNotExist

Esta exceção refere-se a, no momento de log in, ser introduzido um e-mail não associado a nenhuma conta.

### 2.2.5. exInvalidNIF

Esta exceção refere-se a, no momento de criação de uma conta, ser introduzido um NIF com um número de caracteres diferente de 9.

### 2.2.6. exLojaAlreadyExists

Esta exceção refere-se a, no momento de criação de uma conta para uma loja, ser introduzido o nome de uma loja já existente.

### 2.2.7. exNegativeValues

Esta exceção refere-se à introdução de valores negativos ou iguais a 0 para dados como o raio e o preço.

### 2.2.8. exPEAlreadyExists

Esta exceção refere-se a, no momento de criação de uma conta para uma empresa transportadora / voluntário, ser introduzido o nome de uma empresa transportadora / voluntário já existente.

### 2.2.9. exPENotAvailable

Esta exceção refere-se à tentativa de escolher uma plataforma de entrega indisponível.

### 2.2.10. exPWIncorrect

Esta exceção refere-se a, no momento de log in, ser introduzido uma palavra passe incorreta.

### 2.2.11. exEncomendaDoesNotExist

Esta exceção refere-se a, no momento de log in, ser introduzido um código de encomenda não existente.

### 2.2.12. exLojaDoesNotExist

Esta exceção refere-se a, no momento de log in, ser introduzido um código de loja não associado a nenhuma conta.

## 2.3. Classes de Comparadores

O nosso grupo resolveu criar classes de comparadores de forma a organizar listas de *Utilizadores* e *PlataformaEntrega*.

### 2.3.1. cMelhorClassificada

Esta classe compara a classificação média numa lista de *PlataformaEntrega* de modo a entregar a empresa transportadora melhor classificada enquanto o utilizador escolhe a empresa transportadora que pretende utilizar para entregar a sua encomenda.

### 2.3.2. cMelhorCusto

Esta classe compara a classificação média numa lista de *PlataformaEntrega* de modo a entregar a empresa transportadora mais barata enquanto o utilizador escolhe a empresa transportadora que pretende utilizar para entregar a sua encomenda.

### 2.3.3. cTopPE

Esta classe compara o número de encomendas entregues numa lista *PlataformaEntrega* de modo a entregar as 10 empresas transportadoras/ voluntários com o maior número de encomendas entregues.

#### 2.3.4. cTopU

Esta classe compara o número de encomendas entregues numa lista *Utilizadores* de modo a entregar as 10 utilizadores com o maior número de encomendas entregues.

### 3. Interação com a aplicação

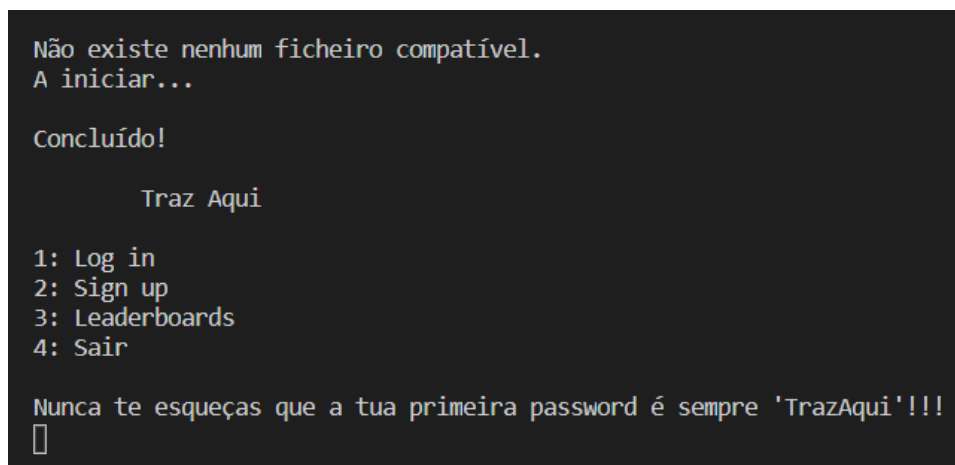
Neste ponto, mostramos como interagir corretamente com a aplicação, com a ajuda de capturas de ecrã da mesma.

Para facilitar esta demonstração, utilizamos o ficheiro de log fornecido.

Nós supomos que o domínio na internet “trazaqui.com” existe e que pode ser utilizado para criar endereços de e-mail. Assim sendo, quando lemos os dados do ficheiro log, interpretamos o e-mail como sendo [código do utilizador] + “@trazaqui.com”. Ou seja, um utilizador poderia escolher fazer log in com o seu código ou com este e-mail gerado por nós.

Ainda em relação à leitura do ficheiro de log que nos foi dado, como não temos lá palavras-passes, é assumido por defeito que a palavra passe é “TrazAqui”. De qualquer forma, é possível a sua alteração a qualquer momento depois de ter efetuado o log in.

#### 3.1. Menu Inicial



```
Não existe nenhum ficheiro compatível.  
A iniciar...  
  
Concluído!  
  
Traz Aqui  
  
1: Log in  
2: Sign up  
3: Leaderboards  
4: Sair  
  
Nunca te esqueças que a tua primeira password é sempre 'TrazAqui'!!!  
□
```

Este é o primeiro menu que aparece no ecrã depois de abrir a aplicação. Note-se que a mensagem “Não existe nenhum ficheiro compatível” refere-se ao facto de não existir um ficheiro object associado à aplicação na altura em que a abrimos.

## 3.2. Sign Up

```
1: Utilizador
2: Loja
3: Transportadora
4: Voluntário
5: Voltar atrás
```

Este é o menu de sign up, onde podemos escolher criar uma conta para um novo utilizador, loja, empresa transportadora e voluntário. Para facilitar a implementação do sistema de sign up/ log in, não é pedida nenhuma palavra-passe durante o sign up, que fica definida como “TrazAqui” para qualquer tipo de utilizador (mas poderá, logicamente, ser alterada).

### 3.2.1. Sign Up – Utilizador

```
Nome
Pedro Martins

Código de utilizador
pedrom

Email
pedromartins@hotmail.com

Localização X
15

Localização Y
16.5
```

Aqui podemos observar o menu de sign up de um *Utilizador*. O utilizador em questão terá que preencher obrigatoriamente todos os campos com atenção, em específico, ao código de utilizador e ao e-mail que não podem ser iguais a outros pertencentes a algum utilizador.

### 3.2.2. Sign Up – Loja

```
Nome
Pingo Doce

Localização X
13

Localização Y
13
```

Neste caso, o sign up é mais básico, tendo apenas a escolha de um nome de loja e definir a sua localização. O código de loja fica definido a partir do nome da loja mas retirando os espaços. Por exemplo, a loja com o nome “Pingo Doce” ficará com o código “PingoDoce”.

### 3.2.3. Sign Up – Transportadora / Voluntário

```
Nome
Transportadora Portuguesa

NIF
579804576

Raio
15

Preço por km
10

Transporte de medicamentos (Yes/No)
Yes

Localização X
6

Localização Y
600
```

O sign up das empresas transportadoras funciona da mesma forma que o das lojas na atribuição do código. Além disso é necessário também apresentar um NIF, preço por quilómetro, raio e confirmar se o transporte de medicamentos é possível.

```
Nome
Aurélio Aragão

Raio
460

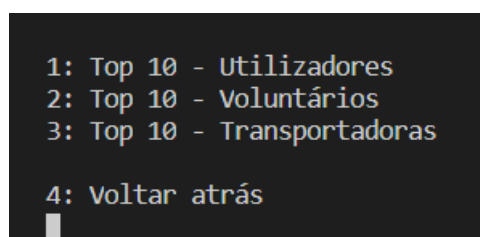
Transporte de medicamentos (Yes/No)
No

Localização X
230

Localização Y
245
```

No caso dos voluntários, é muito idêntico mas é ligeiramente mais simples pelo facto de não existir um NIF, nem nenhum valor a ser cobrado por quilómetro.

### 3.3. Leaderboards



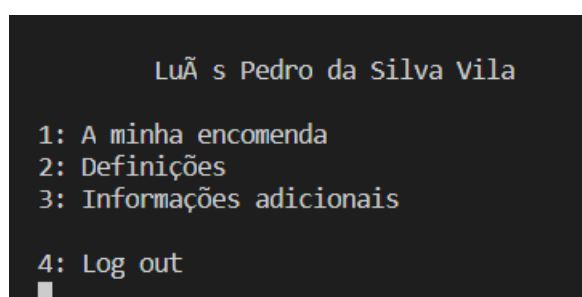
O menu de leaderboards tem 3 opções. A primeira mostra os 10 utilizadores com mais encomendas efetuadas, a segunda mostra os 10 voluntários com mais encomendas entregues e a terceira mostra as 10 transportadoras com mais encomendas entregues.

### 3.4. Log In

O menu principal de log in tem um aspeto muito idêntico do menu de sign up, por isso vamos focar a nossa atenção em cada um dos diferentes menus dentro deste e explicar o funcionamento do resto do programa para cada um deles.

### 3.5. Utilizador

#### 3.5.1. Utilizador – Menu Principal



O menu principal do utilizador tem este aspeto. Aqui o utilizador poderá escolher entre três opções principais.

### 3.5.2. Utilizador – A minha encomenda

```
1: Ver estado da encomenda
2: Inserir pedido de encomenda
3: Pedir transporte
4: Classificar o serviço de transporte
5: Voltar atrás
```

Aqui, o utilizador poderá realizar todas as opções relacionadas com as encomendas. Vamos agora explicar em pormenor o que cada uma destas opções faz.

#### 1) Ver estado da encomenda

```
Não inseriu nenhuma encomenda.

Insira um caracter para voltar atrás.
```

Este é o aspeto deste menu caso ainda não tenhamos escolhido a opção 2. Portanto, antes de explicar o funcionamento da opção 2, vamos apenas inserir uma encomenda para demonstrar o aspeto deste menu depois dessa ação.

```
Tempo estimado de espera
    68 minutos

Custo da encomenda
    1004.0 €

Insira um caracter para voltar atrás.
```

Depois de inserir uma encomenda, o menu do estado da encomenda fica com este aspeto. O custo da encomenda refere-se ao custo dos produtos da encomenda (e não do seu transporte). Esse valor é calculado multiplicando o custo unitário de cada produto pela sua quantidade.

#### 2) Inserir pedido de encomenda

```
Código da encomenda
e1014

Código da loja
18

A sua encomenda contém medicamentos? (Yes/No)
No
```



Para inserir uma encomenda, nós pedimos que o utilizador verifique o código da loja e se a encomenda inclui medicamentos ou não.

### 3) Pedir transporte

```
1: Entrega realizada por transportadora
2: Entrega realizada por voluntário
3: Voltar atrás
```

Quando o utilizador escolhe a opção de pedir o transporte da encomenda é-lhe dada a opção de escolher uma transportadora ou um voluntário para realizar a sua entrega. Inicialmente, vamos escolher uma transportadora.

```
Transportadoras disponíveis

Código: t51
Nome: SÁBIA ATITUDE - CONSTRUÇÃO•ES
Transporte de medicamentos: false
Classificação: 0.0
Preço para a sua entrega: 1234.0

Código: t31
Nome: ZENITHINDEX
Transporte de medicamentos: false
Classificação: 0.0
Preço para a sua entrega: 586.0

Código: t24
Nome: TORRESTIR TRANSITÁRIOS LDA
Transporte de medicamentos: false
Classificação: 0.0
Preço para a sua entrega: 457.0

Código: t26
Nome: TERCARGO - TRANSITÁRIOS
Transporte de medicamentos: false
Classificação: 0.0
Preço para a sua entrega: 118.0

Recomendações para si
  Mais barata: t26
  Melhor classificada: t26

Código da transportadora
```

Aqui é apresentada ao utilizador uma lista de transportadoras disponíveis para a entrega da encomenda. É-lhe também apresentado recomendações: transportadora mais barata e transportadora melhor classificada. Para demonstrar a parte seguinte vamos escolher a transportadora t26, por exemplo.

```
Tempo estimado de entrega
  204 minutos

Custo total de entrega
  118.0 €

Obrigado pela sua escolha!!

Insira um caracter para voltar ao menu de encomenda.
```

Neste momento, o utilizador tem acesso ao tempo estimado da entrega da encomenda, o seu custo.

O tempo estimado é calculado usando a seguinte fórmula:

$$TT = TL + TU$$

$$TL_{aux} = dL * 60 / ((60 \text{ (para voluntários) ou } 65 \text{ (para transportadoras)}) * M)$$

$$TL = \text{round}(TL_{aux} + 0.2 * \text{peso})$$

$$TU_{aux} = dU * 60 / ((60 \text{ (para voluntários) ou } 65 \text{ (para transportadoras)}) * M)$$

$$TU = \text{round}(TU_{aux} + 0.2 * \text{peso})$$

sendo que:

TT – tempo total

TL – tempo estimado entre a localização da transportadora e da loja

dL – distância entre a localização da transportadora e da loja

TU – tempo estimado entre loja e localização do utilizador

dU – distância entre loja e localização do utilizador

M – multiplicador relacionado com as condições atmosféricas:

1 para a primavera

1.1 para o verão

1.25 para o outono

1.5 para o inverno

O custo total é calculado usando a seguinte fórmula:

$$CT = d * p * (1 + \text{peso}/300)$$

sendo que:

CT – custo total

d – distância entre a localização da transportadora e a loja + distância entre loja e localização do utilizador

p – preço por quilómetro

#### 4) Classificar o serviço de transporte

Avalie o serviço de 1 a 10  
10

Aqui, o utilizador pode classificar de 1 a 10 o serviço de transporte.

### 3.5.3. Utilizador – Definições

```
1: Localização atual
2: Histórico de encomendas realizadas
3: Alterar email
4: Alterar password
5: Alterar localização

Aconselhamos a mudar regularmente a sua password.

6: Voltar atrás
```

No menu de definições do utilizador, o utilizador poderá alterar os seus dados como entender e pode também verificar a sua localização atual e verificar o histórico de encomendas que lhe foram entregues.

### 3.5.4. Utilizador – Informações adicionais

```
1: Registo das encomendas de um voluntário/transportadora
2: Voltar atrás
```

```
1: Até à data
2: Num certo período de tempo
3: Voltar atrás
```

Na primeira captura de ecrã podemos escolher a única opção disponível, que é a opção de ver o histórico de encomendas entregues de uma certa *PlataformaEntrega*. Podemos ainda ver esse histórico até ao momento atual, ou podemos escolher um intervalo de tempo.

```
Histórico de entregas realizadas:

Código da encomenda: e1014
Código da loja: l8
Código do utilizador: u78
Peso total: 98.82324
Data: 2020-06-11T10:27:13.438981300
Produtos: 12

Insira um caracter para voltar ao menu principal.
```

Independentemente de escolher a primeira ou a segunda opção, o ecrã resultante terá um aspeto semelhante a este.

## 3.6. Loja

### 3.6.1. Loja – Menu Principal

```
Primark

1: Número de pessoas atualmente na fila
2: Registo de encomendas entregues até à data
3: Registo de encomendas entregues num certo período de tempo
4: Alterar password

5: Log out
```

Neste menu, é possível realizar operações referentes à loja. Iremos demonstrar apenas a opção 1 e a opção 2, sendo que opção 3 é semelhante à anterior e a 4 baseia-se apenas na alteração da palavra-passe.

### 3.6.2. Loja – Número de pessoas atualmente na fila

```
Neste momento estão 0 pessoas na fila.

Insira um caracter para voltar para o menu principal.
```

Aqui, é possível observar o número de voluntários/transportadoras que se encontram na fila para receber a encomenda que precisam de entregar. Como não é efetuada a simulação do tempo e, na prática, a entrega da encomenda é instantânea, a fila irá sempre ter 0 pessoas.

### 3.6.3. Loja – Registo de encomendas entregues

```
Ainda não realizou qualquer encomenda.

Insira um caracter para voltar ao menu principal.
```

Como a loja escolhida (Primark) ainda não realizou nenhuma encomenda, o seu histórico está vazio. Por outro lado, a loja Punt Roma tem o histórico da encomenda realizada anteriormente, como podemos ver na captura de ecrã a seguir:

```
Histórico de encomendas realizadas:

Código da encomenda: e1014
Código do utilizador: u78
Peso total: 98.82324
Data: 2020-06-11T10:27:13.438981300
Produtos: 12

Insira um caracter para voltar ao menu principal.
```

## 3.7. Transportadora

### 3.7.1. Transportadora – Menu Principal

```
TERCARGO - TRANSITÁRIOS

1: Total faturado até à data
2: Total faturado num certo período
3: Classificação atual da empresa
4: Registo de entregas realizadas até à data
5: Registo de entregas realizadas num certo período de tempo
6: Definições

7: Log out
```

Neste menu principal, conseguimos ter acesso a funcionalidades relacionadas com as empresas transportadoras.

### 3.7.2. Transportadora – Total faturado

```
O total faturado é de 136.0.

Insira um caracter para voltar ao menu principal.
```

Escolhendo a opção 1 ou 2 do menu principal, a transportadora terá eventualmente acesso a um ecrã deste género, onde é indicado o dinheiro total ganho até à data, ou durante um intervalo de tempo.

### 3.7.3. Transportadora – Classificação atual da empresa

```
A classificação atual da sua empresa é de 10.0.

Insira um caracter para voltar ao menu principal.
```

Como numa das demonstrações, avaliamos a entrega da encomenda com 10 pontos, agora podemos observar que a média das classificações é realmente de 10.0.

### 3.7.4. Transportadora – Registo de entregas realizadas

```
Histórico de entregas realizadas:

Código da encomenda: e1014
Código da loja: l8
Código do utilizador: u78
Peso total: 98.82324
Data: 2020-06-11T11:13:40.263204200
Produtos: 12

Insira um caracter para voltar ao menu principal.
```

Tanto usando a opção 4 ou a opção 5, teremos acesso a um histórico de entregas realizadas deste género.

### 3.7.5. Transportadora – Definições

```
1: Alterar password
2: Alterar raio
3: Alterar capacidade da empresa
4: Alterar preço por km
5: Validar o transporte de medicamentos

6: Voltar atrás
```

Nas definições da transportadora, podem ser alterados os seus dados como for pretendido.

## 3.8. Voluntário

### 3.8.1. Voluntário – Menu Principal

```
Pedro Miguel da Silva Paiva Ant3nio

1: Classificação atual
2: Registo das entregas realizadas até à data
3: Registo das entregas realizadas num certo período de tempo
4: Definições

5: Log out
```

No menu principal do voluntário, temos acesso à sua classificação média, ao histórico das entregas realizadas e às definições.

### 3.8.2. Voluntário – Classificação atual

```
A sua classificação atual é de 0.0.  
Insira um caracter para voltar ao menu principal.
```

Funcionando de uma forma idêntica à transportadora, este menu inclui a classificação média de um voluntário.

### 3.8.3. Voluntário – Registo das entregas realizadas

```
Ainda não realizou qualquer entrega.  
Insira um caracter para voltar ao menu principal.
```

Tal como no caso das transportadoras, escolhendo a opção 2 ou 3 irá eventualmente levar a um ecrã deste género, onde são mostradas as entregas realizadas.

### 3.8.4. Voluntário – Definições

```
1: Alterar password  
2: Alterar raio  
3: Validar o transporte de medicamentos  
4: voltar atrás
```

Nas definições, o voluntário poderá alterar os seus dados como pretender.

## 4. Conclusão

O nosso grupo considera que o projeto foi desenvolvido de forma a implementar as funções pedidas. Em termos de código e de apresentação gráfica acreditamos ter feito um bom trabalho em desenvolver métodos úteis e eficientes. Realmente o único pormenor que gostávamos (e numa fase inicial estávamos a planear desenvolver) eram *views* para mostrar a aplicação, mas por falta de tempo e até por falta de confortabilidade no desenvolvimento de *views* decidimos que nos devíamos focar em fazer realmente uma aplicação funcional e, na verdade, achamos que o facto de termos desenvolvido a parte gráfica da aplicação com impressões de texto no ecrã não foi um fator negativo, porque fizemos o pudemos para apresentar o texto de uma forma intuitiva e explicativa.

Por outro lado, também acreditamos que mesmo que surgisse alguma dúvida sobre a aplicação ou o seu código, este relatório será capaz de esclarecer isso.