

Universidade do Minho
Programação Orientada aos Objetos

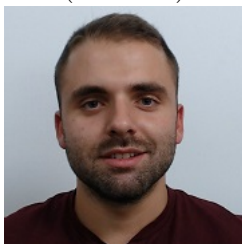


(LCC - 2º Ano)
Trabalho Prático
Relatório

João Silva
(A91638)



Eduardo Pereira
(A70619)



Áureo Benedito
(A76068)



14 de maio de 2023

Conteúdo

1	Introdução	2
2	Descrição do Enunciado Proposto	3
3	Descrição da Arquitetura de Classes	4
4	Descrição da Aplicação	6
4.1	Inicialização de Menu	6
4.2	Atuar como Utilizador	6
4.3	Funcionalidades do Utilizador	7
4.4	Adicionar Artigos	7
4.5	Criar Encomendas	7
4.6	Operações da loja	8
5	Diagrama de classes	9
6	Conclusão	10

Capítulo 1

Introdução

Este relatório debruça-se sobre o desenvolvimento de um sistema de marketplace Vintage, no âmbito da unidade curricular de Programação Orientada aos Objetos.

Este sistema permite aos utilizadores efetuar compra e venda de artigos, novos ou usados. Também tem implementado um conjunto de transportadoras, bem como as encomendas e faturas das respetivas compras. Além deste conjunto de ferramentas, um utilizador consegue saber informações sobre o estado dos produtos e o uso que um determinado produto teve, entre outra informação que vamos especificar no decorrer deste relatório.

Ao longo deste relatório iremos desenvolver este tema, comentar as escolhas feitas para o resolver, assim como problemas ou dificuldades que encontramos durante a sua realização.

Capítulo 2

Descrição do Enunciado Proposto

A aplicação Vintage, tem como objetivo permitir a compra e venda de artigos de vários tipos. Esta aplicação permite ao utilizador, que atua como vendedor e comprador, realizar várias operações, como expor artigos para vender, comprar artigos criando encomendas, entre outras.

Todas as compras e vendas são geridas a partir do sistema. Quando se efetua uma compra, é criada uma encomenda, que pode conter vários artigos, sendo que estes artigos têm de estar associados à mesma transportadora. Pretende-se também que neste trabalho seja possível criar ou remover artigos, utilizadores, encomendas, transportadoras e faturas. Além disto, será possível salvar o estado atual do programa em ficheiro, de forma a ser possível aceder a esta informação, dispensando o processo de setup da informação quando se volta a iniciar o programa.

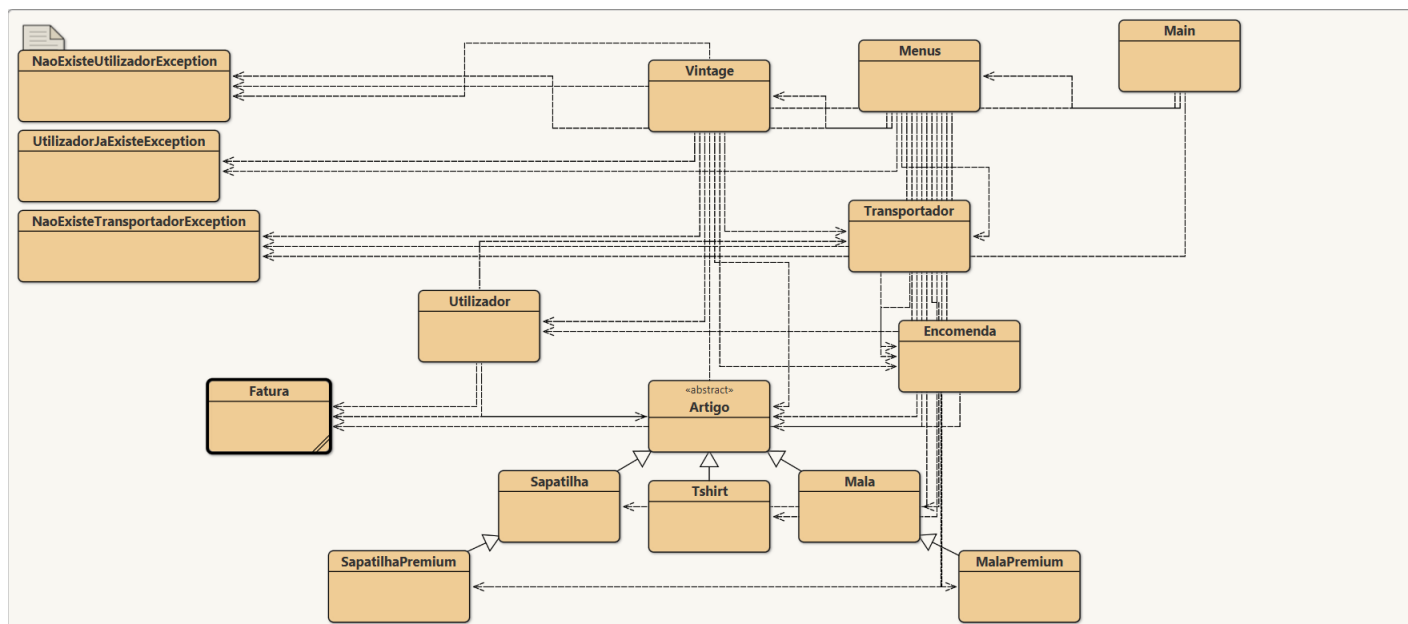


Figura 2.1: Estrutura

Capítulo 3

Descrição da Arquitetura de Classes

Como neste trabalho temos classes que partilham alguma informação, como o código do artigo, o seu estado de utilização, o número de donos anteriores e o seu preço base, decidimos criar a classe abstrata **Artigo**, que tem como suas subclasses:

- **Tshirt**, que contém informação sobre o seu padrão e o tamanho
- **Sapatilha**, que contém informação sobre a cor, tamanho e se tem atacadores ou não
- **Mala**, que contém informação sobre a dimensão e ano de coleção

Tanto a sapatilha, como a mala, têm as subclasses **SapatilhaPremium** e **MalaPremium**, respetivamente. Estas subclasses contém informações sobre o autor da coleção.

Implementamos, também várias classes para facilitar o processo de compra neste serviço, sendo estas:

- **Utilizador**, que tem informação referente ao seu NIF, morada, email e nome. Também tem informações adicionais sobre os artigos que este tem à venda, os seus artigos que se encontram em processo de venda, e o valor total das suas vendas e todas as faturas provenientes das suas compras.
- **Fatura**, que tem alguma informação sobre o utilizador que efetuou a compra e sobre os artigos que foram comprados e respetivos preços. Contém também informação sobre os custos adicionais que se deverão ter em conta, como os preços adicionais dos custos de transporte, e algum tipo de imposto.
- **Encomenda**, que tem informação sobre os artigos nela contidos e a morada e nome do comprador.
- **Transportador**, que terá um nome, informação sobre as suas encomendas, e uma fórmula para calcular o valor de expedição para cada encomenda.
- **Vintage**, que terá informação relativa a todas as classes já referidas, contendo um Map que fará corresponder o número fiscal de um utilizador ao próprio, uma coleção de transportadoras associadas com a loja, e também, uma coleção de códigos já gerados para todos os artigos e utilizadores, de forma a que estes códigos não se repitam.

Criamos a classe **Menus**, que possibilita a interação entre o cliente e o programa. Esta classe oferece a possibilidade de simular a criação de artigos para vender, assim como a criação de utilizadores, transportadoras, encomendas e faturas, entre outras.

Quando inicializada a simulação, é lido de um ficheiro o estado do programa previamente gravado, sendo, depois, possível atuar de dois modos, como um utilizador normal, que pode ser comprador ou vendedor, ou atuar como um administrador do site da loja, que tem possibilidade de visualizar um espetro de informações adicionais. Após realizar todas as operações pretendidas, é gravado num ficheiro de objetos o estado do programa atual, para poder voltar a ser lido novamente.

A classe **Main** possui o método main, que relaciona todo o programa com o respetivo input/output.

Temos também tres classes que tratam exceções:

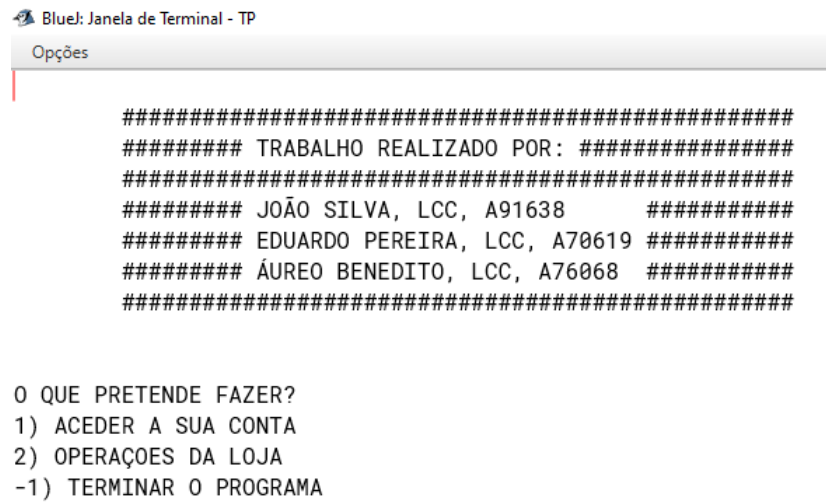
- **UtilizadorJaExisteException** lançada quando já existe um NIF associado a um utilizador da loja, ao tentar criar um utilizador.
- **NaoExisteUtilizadorException** lançada quando, se procura por um utilizador que não pertence à loja.
- **NaoExisteTransportadorException** lançada quando se procura por uma transportadora não associada à loja.

Capítulo 4

Descrição da Aplicação

4.1 Inicialização de Menu

Através de **Menu** conseguimos inicializar a aplicação:



```
BlueJ: Janela de Terminal - TP
Opções

#####
##### TRABALHO REALIZADO POR: #####
#####
##### JOÃO SILVA, LCC, A91638 #####
##### EDUARDO PEREIRA, LCC, A70619 #####
##### ÁUREO BENEDITO, LCC, A76068 #####
#####

O QUE PRETENDE FAZER?
1) ACEDER A SUA CONTA
2) OPERAÇÕES DA LOJA
-1) TERMINAR O PROGRAMA
```

Figura 4.1: Menu Inicial

Tal como se pode ver na **Figura 4.1**, temos a possibilidade de fazer login e atuar como um utilizador em que podemos expor artigos para venda e criar encomendas, ou de atuar como a própria loja, e pedir informações a que apenas a própria terá acesso, como o utilizador que faturou/gastou mais em vendas/compras na loja, a transportadora que faturou mais com encomendas, quanto faturou a loja, entre outros.

4.2 Atuar como Utilizador

Caso já se tenha efetuado um registo anteriormente, basta que seja feito login para entrar.

4.3 Funcionalidades do Utilizador

Seguem ilustradas, todas as opções que o utilizador poderá tomar, quando o login é feito.

```
0 QUE PRETENDE FAZER?
1) ACEDER A SUA CONTA
2) OPERAÇÕES DA LOJA
-1) TERMINAR O PROGRAMA
1
INSIRA O SEU NIF
4
INSIRA O SEU EMAIL
mau
        acesso permitido

        Que Operacao Pretende Realizar?

1----Criar artigo
2----Remover artigo
3----Criar encomenda
4----Adicionar artigo a encomenda
5----Remover artigo de encomenda
6----Finalizar encomenda
7----Ver os meus artigos expostos para venda
8----Ver os meus artigos em processo de venda
9----Ver os meus artigos vendidos
10---Ver os artigos que comprei
11---Ver as minhas encomendas
12---Ver as minhas faturas
-1---Fazer logout
```

Figura 4.2: Login

4.4 Adicionar Artigos

1. Para adicionar um artigo a uma encomenda, será preciso primeiro escolher uma transportadora para associar a este artigo.
2. Escolhida a transportadora, escolhe-se o tipo de artigo a expor para venda.

4.5 Criar Encomendas

O *Utilizador*, pode inserir um pedido de encomenda.

1. O primeiro passo será a escolha de uma transportadora para a encomenda
2. De seguida, são listados todos os artigos associados à transportadora escolhida. Basta carregar na tecla (1) para adicionar um artigo à encomenda. No final do processo, a encomenda é criada em estado pendente
3. Existe a possibilidade de adicionar e remover artigos à encomenda, enquanto esta estiver pendente.
4. Por último, basta apenas finalizar a encomenda.
5. Para as encomendas serem expedidas e entregues, é necessário aceder às funcionalidades da loja, escolhendo uma transportadora e uma encomenda sua, para ser expedida ou entregue.
6. Quando a encomenda é entregue pela transportadora, o Utilizador receberá uma fatura relativa à compra efetuada.


```
0 QUE PRETENDE FAZER?
1) ACEDER A SUA CONTA
2) OPERAÇÕES DA LOJA
-1) TERMINAR O PROGRAMA
2
    Que Operacao Pretende Realizar?

1---Criar utilizador
2---Remover utilizador
3---Imprimir utilizadores
4---Criar transportador
5---Imprimir transportadores
6---Transportador expede encomenda
7---Transportador entrega encomenda
8---Imprime encomendas de uma transportadora
9---Utilizador que mais vendeu
10--Utilizador que mais gastou em compras da loja
11--Transportadora que mais faturou em encomendas da loja
12--Valor faturado pela loja
13--Informacoes adicionais sobre a loja
-1--Voltar atras
```

Figura 4.3: Operações da loja

4.6 Operações da loja

Na imagem acima podemos ver todas as operações sobre a própria loja que podemos efetuar, como por exemplo:

- Criar e remover utilizadores.
- Criar transportadores.
- Escolher um transportadora para expedir ou entregar uma encomenda.
- E ainda temos acesso a varias informações adicionais, como o utilizador que mais faturou, o utilizador que mais gastou em compras na loja, a transportadora que mais faturou, o valor total faturado pela loja, entre outros.

Diagrama de classes

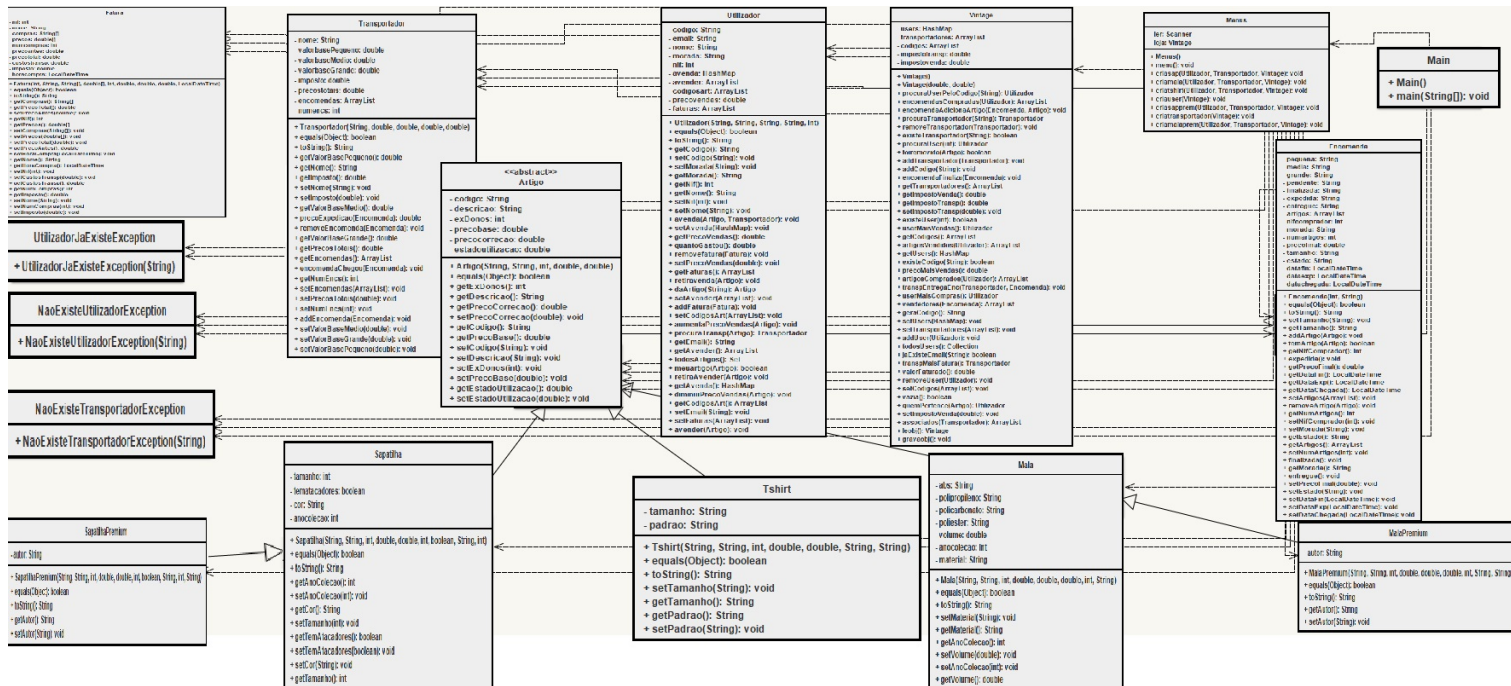


Figura 5.1: Diagrama de classes

Capítulo 6

Conclusão

Tendo em conta o trabalho desenvolvido, o respeito pelo encapsulamento e os princípios básicos da programação orientada aos objetos, o grupo encontra-se satisfeito por ter conseguido cumprir os requisitos mínimos, como criar e remover utilizadores, artigos, encomendas, e transportadoras. Conseguimos também implementar bastantes funcionalidades avançadas, como realizar estatísticas sobre transportadoras, utilizadores e também da própria loja.

No entanto, reconhecemos que poderiam ter sido implementadas mais funcionalidades no programa, como a devolução de encomendas, e a utilização de mais exceções, que não foi realizado pois, com relativamente pouco tempo até à entrega do projeto, ficamos um pouco reticentes em alterar demasiado a estrutura do trabalho.