



UNIVERSIDADE DO MINHO
Departamento de Informática

PROGRAMAÇÃO ORIENTADA AOS OBJETOS

Trabalho Prático

Realizado por:



João Coelho (a100596)



José Rodrigues (a100692)



Duarte Araújo (a100750)

May 14, 2023
Ano Letivo 2022/23

Índice

1	Introdução	3
2	Classes	3
2.1	Main	3
2.2	Vintage	3
2.3	Artigos	4
2.4	Sapatilhas, Malas e TShirt	4
2.5	SapatilhasPremium e MalasPremium	5
2.6	Utilizador	5
2.7	Encomenda	6
2.8	Transportadora e TransportadoraPremium	6
2.9	VintageController, AdminController e UtilizadorController	6
2.10	Menu, MainView e UtilizadorView	7
2.11	Codigos	7
2.12	EstadoPrograma	7
2.13	VintageException e Read_Scanner	7
3	Estatísticas	9
4	Aplicação	9
4.1	EXIT	9
4.2	Selecionar View	10
4.3	View Admin	10
4.4	View Utilizador	11
5	Conclusão	13

1 Introdução

No âmbito da cadeira de Programação Orientada a Objetos foi-nos pedido que desenvolvêssemos um programa Vintage que simula um Marketplace de artigos, ou seja, permite a venda e a compra de diversos produtos.

Para além disso, o programa deve permitir o registro de utilizadores, que serão responsáveis pelo movimento de artigos, sendo que estes podem ser compradores, vendedores ou ambos. Devemos realçar que o programa é responsável por gerir todas as vendas e compras realizadas.

Por fim, as compras realizadas são organizadas em encomendas que são posteriormente expedidas por diversas empresas de transporte e, por cada encomenda, a Vintage cobra uma taxa de garantia de serviço ao vendedor, garantindo a segurança e satisfação dos compradores caso as encomendas não sejam entregues.

2 Classes

Para que o programa fosse capaz de cumprir todos os requisitos numerados no enunciado do trabalho foram criadas diversas classes. Cada uma com funções específicas que, no seu conjunto, permitem o bom funcionamento do programa.

De seguida, cada uma das classes será explicada detalhadamente e serão apresentados os seus atributos e o porquê de os termos definido. É importante realçar que não serão objeto de discussão os requisitos auto-explicativos de cada classe, ou seja, requisitos como **String nome** que, na nossa opinião, não necessitam de justificação.

2.1 Main

A classe main é responsável pela criação de todos os componentes necessários para o funcionamento do programa e inicialização da execução do mesmo. Desta forma a classe não possui nenhum atributo.

2.2 Vintage

Esta classe é a classe principal do programa e guarda todas as informações da aplicação. Os atributos desta classe são:

```
private LocalDate currentDate;  
private double rendeu;  
private Map<String,Utilizador> utilizadores;  
private Map<String,Artigos> artigos;  
private Map<String,Artigos> artigosDisponiveis;  
private Map<String,Transportadora> transportadoras;
```

Atributos da classe Vintage

O Map <String,Artigos > artigosDisponiveis guarda todos os artigos que ainda estão à venda.

2.3 Artigos

A classe artigos define os dados que estarão associados a um produto no contexto do nosso trabalho. Os atributos presentes nesta classes são os seguintes:

```
public enum Estado{
    Novo,
    Usado
}
private Estado estado;
private String danos;
private int nDonos;
private String descricao;
private String marca;
private String codigo;
private String user_id;
private Transportadora transportadora;
private double preco;
private double preco_desconto;
private LocalDate data_venda;
```

Atributos da classe Artigos

Como foi referido anteriormente, só serão apresentados os atributos que consideramos pertinentes que, nesta classe, só existe um, a **LocalDate data_venda**, que, apesar do nome ser auto-explicativo, não faz parte dos atributos base dados no enunciado. Este atributo é utilizado para auxiliar a execução de uma das estatísticas, que serão explicadas futuramente, e guarda a data em que o artigo foi vendido.

2.4 Sapatilhas, Malas e TShirt

Estas 3 classes herdam a classe Artigos apresentada previamente e representam o tipo de artigos que existem no sistema. Todas têm acesso à totalidade dos atributos dos artigos devido à herança, no entanto, cada uma delas ainda possui atributos únicos.

```
public enum Dimensao {
    Pequena,
    Media,
    Grande
}

public enum Material {
    Polipiel,
    Couro,
    Tecido
}

private Dimensao dimensao;
private Material material;
private LocalDate date;
private double descontoMaterial;
```

Atributos da classe Malas

```
public enum Tamanho {
    S,
    M,
    L,
    XL
}

public enum Padrao {
    Liso,
    Riscas,
    Palmeiras
}

private Tamanho tamanho;
private Padrao padrao;
private LocalDate date;
```

Atributos da classe TShirt

```
private int tamanho;
private int atacador;
private String cor;
private LocalDate date;
```

Atributos da classe Sapatilhas

Nas três classes, a **LocalDate date** corresponde à data da coleção do produto. Para além disso, na classe Malas o **double descontoMaterial** corresponde ao desconto associado ao material da mala (desconto 5% - $\text{descontoMaterial} = 0.05$).

2.5 SapatilhasPremium e MalasPremium

Classes responsáveis por criar a noção de premium para os artigos no nosso projeto e que herdam a sua versão não premium, ou seja, SapatilhasPremium herda Sapatilhas e MalasPremium herda Malas. Nenhuma destas classes possui atributos próprios, no entanto, devido à herança, têm acesso à totalidade dos atributos da classe que herdam.

Assim, é possível chegar ao seguinte diagrama de classes para os artigos:

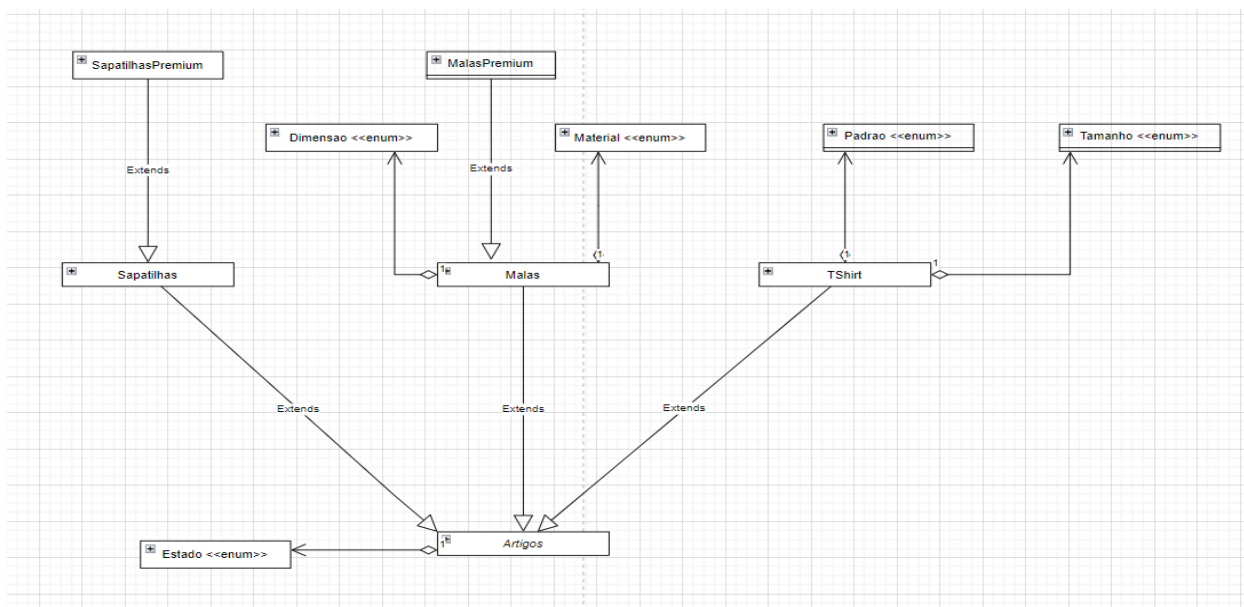


Diagrama de classes dos Artigos

2.6 Utilizador

A classe utilizador define os dados associados aos vendedores e compradores da Vintage. Os atributos da classe são:

```

private int nif;
private String id,nome,morada,email;
private double preco_vendidos;
private Map <String, Artigos> venda,vendeu;
private List <Artigos> carrinho;
private Map <String, Encomenda> encomendas;
private List <String> faturas;

```

Atributos da classe Utilizador

Os `Map <String, Artigos>` `venda` e `vendeu` guardam todos os produtos que o utilizador tem à venda ou já vendeu e, por sua vez, o `double preco_vendidos` retém a quantidade de dinheiro ganha pelo utilizador devido à venda de produtos.

2.7 Encomenda

Esta classe define o conceito de encomenda na Vintage, bem como define os dados que terão de estar associados a cada encomenda. Os atributos presentes na classe encomenda são os seguintes:

```
public enum Dimensao {
    Pequena,
    Media,
    Grande
}

public enum Estado {
    Pendente,
    Finalizada,
    Expedida
}

private String codigo;
private double preco;
private Dimensao dimensao;
private Estado estado;
private LocalDate data;
private List<Artigos> lista;
private Transportadora transportadora;
```

Atributos da classe Encomenda

A `LocalDate data` guarda a data em que a encomenda é criada.

2.8 Transportadora e TransportadoraPremium

A classe `Transportadora` é responsável por definir todos os dados que têm de estar associados a uma transportadora na Vintage, enquanto a classe `TransportadoraPremium` cria a noção de premium para transportadoras e herda a classe `Transportadora`, logo tem acesso à totalidade dos seus atributos, no entanto não tem atributos próprios. Os atributos da classe `Transportadora` são:

```
private double margemLucro, imposto, valorBasePeq, valorBaseMed, valorBaseGra, totalObtido;
private String transportadora;
```

Atributos da classe Transportadora

O `double imposto` é um valor fixo, ou seja, nunca será alterado, e corresponde a 0.23 (23% de imposto). Para além disso, o `double totalObtido` será utilizado para auxiliar o funcionamento de uma das estatísticas e guarda o rendimento total da transportadora.

2.9 VintageController, AdminController e UtilizadorController

Estas classes são responsáveis por criar a ponte entre as Views e o model (arquitetura MVC). Os atributos destas classes são:

```
private Vintage v;
private UtilizadorController user;
private AdminController admin;
```

VintageController

```
private Vintage v;
```

AdminController

```
private Vintage v;
private Utilizador u;
```

UtilizadorController

Atributos das classes Controller

A VintageController, como é o controller principal, tem como atributo a Vintage e os outros dois controllers e, por sua vez, o UtilizadorController, para além da vintage, tem ainda um **Utilizador u**. Este utilizador corresponde ao user que fez login na aplicação.

2.10 Menu, MainView e UtilizadorView

Estas classes, tal como os Controllers, também pertencem à arquitetura MVC e são responsáveis pelas Views do projeto, ou seja, pela parte visual (print dos menus). A classe Menu não tem nenhum atributo, no entanto, as outras duas classes têm:

```
private VintageController v;
```

MainView

```
private UtilizadorController user;
```

UtilizadorView

Atributos das classes View

2.11 Codigos

Esta classe é utilizada para a criação aleatória de códigos que serão utilizados como identificadores por várias outras classes e não possuiu atributos.

2.12 EstadoPrograma

Classes responsável por guardar o estado do programa quando a execução é terminada e, quando o programa volta a ser iniciado, carregar o mesmo para que possa ser possível ter acesso a todos os dados criados previamente no programa. A classe não possui nenhum atributo.

2.13 VintageException e Read_Scanner

Estas duas classes foram criadas com o intuito de captar exceções e retornar as respetivas mensagens de erro e nenhuma delas possui atributos.

Com todas as classes definidas chega-se ao seguinte diagrama de classes do projeto:

3 Estatísticas

Neste projeto foi-nos pedido que o nosso programa fosse capaz de responder a um conjunto de queries que efetuavam estatísticas sobre o estado do programa. De seguida, serão apresentadas as queries fornecidas e as respetivas técnicas utilizadas pelo nosso grupo para as resolver.

- **Qual é o vendedor que mais faturou num período ou desde sempre** - Para resolver esta query adicionámos um atributo à classe Artigos, **LocalDate data_venda**, já apresentado, com o objetivo de nos permitir saber, caso seja dado um intervalo de tempo, se o artigo foi comprado dentro do intervalo ou não. Depois, para saber qual o vendedor que mais faturou, para todos os utilizadores, percorremos o seu **map vendeu** e calculámos o total da soma dos preços (depois dos descontos) dos produtos já vendidos e devolvemos como resultado da query o email daquele que apresentar a soma total mais elevada. É importante referir que, ao percorrer o map, é feita a verificação, caso seja necessário, se o produto foi vendido dentro do intervalo pretendido.
- **Qual o transportador com maior volume de faturação** - À semelhança da query anterior, para facilitar a resolução, foi adicionado um atributo à classe Transportadora, **double totalObtido**, que acumula o rendimento total da transportadora. Assim, quando as encomendas são criadas, é adicionado ao totalObtido da transportadora responsável por entregá-las o rendimento obtido. Finalmente, para saber qual transportadora mais faturou, só é necessário percorrer o **map transportadoras** da classe Vintage e verificar qual a transportadora com o totalObtido mais elevado.
- **Listar as encomendas emitidas por um vendedor** - Para resolver esta query é pedido que seja fornecido o utilizador que será alvo de procura. De seguida, é percorrido o **map utilizadores** da classe Vintage e, para cada utilizador, ainda é percorrido o seu **map encomendas**. Depois, verifica-se se alguma encomenda contém algum artigo do utilizador requisitado e, se tiver, essa encomenda será adicionada a uma lista de encomendas que, após terem sido percorridas todas as encomendas de todos os utilizadores, será devolvida como resultado.
- **Fornecer uma ordenação dos maiores compradores/vendedores do sistema durante um período a determinar** - Com o objetivo de obter esta estatística, iteramos por todos os utilizadores, obtendo o total de venda/compra no período de tempo pedido inicialmente. A cada iteração, o utilizador é introduzido numa *TreeMap* para já ficarem ordenados e, por fim, é realizada a impressão da lista de utilizadores com ordem descendente de valores.
- **Determinar quanto dinheiro ganhou o Vintage no seu funcionamento** - O dinheiro que a Vintage ganha no seu funcionamento corresponde à taxa de satisfação de serviço aplicada durante o cálculo do preço da encomenda. Então, adicionámos um atributo à classe Vintage, **double rendeu**, e, quando uma encomenda é criada, são percorridos os artigos da mesma e calcula-se a soma total das taxas de satisfação na encomenda. De seguida, é adicionado esse valor ao atributo criado. Assim, quando se pede para responder a esta query, é retornado o valor do atributo **rendeu**.

4 Aplicação

Com toda a arquitetura definida torna-se possível visualizar a aplicação Vintage que o nosso grupo foi capaz de criar.

4.1 EXIT

É importante realçar que a opção EXIT será ignorada ao longo da explicação da aplicação, já que a sua função é a mesma em todos os menus. Quando é selecionada a opção EXIT, o menu em que se encontra é terminado e a aplicação mostra o menu antecessor, no entanto, se se encontrar no MenuPrincipal, primeiro menu da aplicação, ao selecionar EXIT o mesmo irá terminar e a aplicação também, para além disso, é nesta fase que o estado do programa é guardado.

4.2 Selecionar View

Primeiramente, o utilizador é deparado com um pequeno menu que permite escolher se pretende entrar no programa como administrador ou utilizador (view admin e view utilizador).

```

-----MENU-----
1 - Utilizador
2 - Admin
0 - EXIT
Choice:

```

Fig. 1: Menu Principal

4.3 View Admin

Se, no menu referido previamente, seleccionar a opção **administrador(admin)**, será apresentado um novo menu com 3 opções.

```

-----MENU-----
1 - Gerir Transportadoras
2 - Estatística
3 - Avançar Tempo
0 - EXIT
Choice:

```

Fig. 2: Menu Admin

Caso escolha a opção **Gerir Transportadoras** será mostrado um outro menu.

```

-----MENU-----
1 - Ver transportadoras
2 - Adicionar Transportadora
3 - Remover Transportadora
4 - Alterar transportadora
0 - EXIT
Choice: |

```

Fig. 3: Menu Transportadoras

A primeira opção, caso seja seleccionada, irá mostrar todas as transportadoras presentes no sistema no momento da sua seleção, por sua vez, ao optar pela segunda opção, irá ser pedido que dê os dados necessários para a criação de uma nova transportadora, que por sua vez vai ser adicionada às transportadoras disponíveis na aplicação. Para além disso, a terceira e a quarta opção, quando são escolhidas, listam as transportadoras registadas no programa e, no caso da opção **Remover Transportadora**, será pedido que indique que transportadora pretende remover, enquanto na opção **Alterar Transportadora** será pedido que indique que transportadora pretende alterar e que dê os novos valores a considerar.

Voltando ao menu do admin, restam as opções **Estatística** e **Avançar Tempo**. Ao seleccionar a primeira será apresentado um outro menu.

```

-----MENU-----
1 - Utilizador que mais rendeu
2 - Transportadora que mais faturou
3 - Encomendas emitidas por um vendedor
4 - Maiores compradores/vendedores do sistema
5 - Rendimento do Vintage
0 - EXIT
Choice: |

```

Fig. 4: Menu Estatísticas

Cada opção é responsável por devolver o resultado de uma das 5 queries apresentadas no enunciado e que já foram explicadas previamente na seção das estatísticas.

Finalmente, voltando ao menu da figura 2, ainda falta selecionar a opção **Avançar Tempo** que, como o nome indica, irá pedir ao admin que forneça um inteiro que, por sua vez, corresponderá ao número de dias que o sistema irá saltar.

Assim, os menus e opções exclusivas do administrador estão todas apresentadas e podemos passar para as escolhas que um utilizador tem acesso.

4.4 View Utilizador

No menu da figura 1, se selecionar a opção utilizador, será deparado com um novo menu.

```

-----MENU-----
1 - Register
2 - Login
0 - EXIT
Choice:

```

Fig. 5: Menu Register/Login utilizador

Aqui, poderá escolher registar-se na aplicação, caso ainda não o tenha feito, e serão pedidos os dados necessários para que o registo possa ser finalizado, sendo que, após o processo ter sido concluído, o novo utilizador será adicionado à aplicação. Por outro lado, se optar pela opção **Login**, será requisitado o email utilizado no registo para poder prosseguir na aplicação.

Se tudo correr bem e conseguir concluir o login, será defrontado com um novo menu.

```

-----MENU-----
1 - Comprar Artigos
2 - Vender Artigos
3 - Ver Carrinho
4 - Ver Encomendas
5 - Ver Faturas
0 - EXIT
Choice:

```

Fig. 6: Menu Utilizador

Neste menu, se quiser prosseguir com a opção **Comprar Artigos**, serão apresentados os vários produtos que estão à venda, caso existam, e será pedido que dê o identificador do artigo que pretende comprar. Por sua vez, se seleccionar a opção **Vender Artigos**, irá aparecer mais um menu.

```

-----MENU-----
1 - Sapatilha
2 - Mala
3 - TShirt
0 - EXIT
Choice:

```

Fig. 7: Menu Venda de Produtos

Cada uma das opções deste menu, ao ser seleccionada, será responsável por requerir que o utilizador forneça os dados necessários para poder registar um novo produto na aplicação, ou seja, colocar à venda, incluindo o preço do artigo e a transportadora responsável por entregá-lo ao comprador. Por exemplo, se seleccionar a primeira opção(Sapatilha), terá de dar todos os dados das sapatilhas que pretende vender na aplicação.

De seguida, de volta ao menu do utilizador (figura 6), poderá ainda escolher a opção ver carrinho que, por sua vez, à semelhança da opção anterior, também irá apresentar um novo menu.

```

-----MENU-----
1 - Validar Carrinho
2 - Remover Artigo
0 - EXIT
Choice:

```

Fig. 8: Menu Carrinho

Neste menu poderá seleccionar a opção **Validar Carrinho**, que irá percorrer os artigos que se encontram no carrinho e dividi-los por encomendas, separadas por transportadoras, que serão posteriormente expedidas. Para além disso, ainda há a opção **Remover Artigo** que, ao ser escolhida, irá pedir que dê o identificador do artigo que pretende remover do carrinho.

Mais uma vez, voltamos ao menu da figura 6 onde ainda podem ser seleccionadas as opções **Ver Encomendas** e **Ver Faturas**. Se optar pela primeira será, novamente, deparado com um outro menu.

```

-----MENU-----
1 - Devolver Encomenda
0 - EXIT
Choice: |

```

Fig. 9: Menu Encomenda

Este menu só possui uma única opção, **Devolver Encomenda**, que, ao ser selecionada, irá listar as encomendas que ainda podem ser devolvidas e irá pedir o identificador da encomenda que pretende devolver.

Por fim, de volta ao menu do utilizador (figura 6), ao selecionar a última opção, **Ver Faturas**, são apresentadas todas as faturas associadas ao utilizador, se existirem.

É importante referir que, apesar de não ser visível nas fotos dos menus, sempre que uma exceção é captada, irá ser apresentada uma mensagem de erro. Por exemplo, se inserir um inteiro superior ao número da opção com o índice mais elevado, irá aparecer a seguinte mensagem de erro "INVALID CHOICE!".

Assim, fica concluída a explicação de todas as funcionalidades da aplicação desenvolvida pelo nosso grupo.

5 Conclusão

Este projeto permitiu aprofundar o conhecimento acerca do funcionamento das linguagens orientadas a objetos. Contudo, foram várias as dificuldades enfrentadas e, apesar de muitas terem sido ultrapassadas, algumas demonstraram ser persistentes até a conclusão do trabalho. Uma das mais presentes é o encapsulamento que, mesmo após o trabalho já estar finalizado, ainda temos dúvidas acerca do mesmo e se o nosso trabalho respeita as suas barreiras. No entanto, acreditamos que fizemos um bom trabalho e demonstramos saber trabalhar com este novo tipo de linguagem de programação.