



# Programação Orientada aos Objectos

## Projecto prático

### Grupo 39

14 de maio de 2023



Alexandra Candeias (A89521) Alexandre Martins (A93242) Diogo Barros (A100600)

# Índice

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Arquitetura de Classes</b>	<b>3</b>
2.1	Main . . . . .	4
2.2	Vintage . . . . .	4
2.2.1	Atributos . . . . .	4
2.3	Ui . . . . .	4
2.4	Parser . . . . .	4
2.5	Utilizador . . . . .	5
2.5.1	Atributos . . . . .	5
2.6	Transportadora . . . . .	5
2.6.1	Atributos . . . . .	5
2.7	Encomenda . . . . .	6
2.7.1	Atributos . . . . .	6
2.8	Artigo . . . . .	6
2.8.1	Atributos . . . . .	6
<b>3</b>	<b>Mala</b>	<b>7</b>
3.0.1	Atributos . . . . .	7
3.1	Sapatilha . . . . .	7
3.1.1	Atributos . . . . .	7
3.2	T-shirt . . . . .	7
3.2.1	Atributos . . . . .	7
<b>4</b>	<b>Descrição da Aplicação</b>	<b>8</b>
4.1	Menu Principal . . . . .	8
4.2	Menu Utilizador . . . . .	8
<b>5</b>	<b>Conclusão</b>	<b>11</b>

# 1 Introdução

Na unidade curricular de Programação Orientada a Objetos, foi-nos proposta a implementação de uma aplicação de entrega de encomendas aos utilizadores, por parte de Transportadoras e Utilizadores. Inicialmente, o foco prioritário deste projeto foi o encapsulamento das estruturas de dados por nós utilizadas, respeitando a metodologia da programação orientada aos objetos.

Ao longo do desenvolvimento do projeto, consideramos que o maior desafio foi a implementação de certas funcionalidades na aplicação impostas pelo enunciado, mantendo, ao mesmo tempo, todas as outras funcionais.

## 2 Arquitetura de Classes

De modo a desenvolver a aplicação é necessário organizar dados numa estrutura que, apesar de compacta, seja de rápido acesso. As classes elementares da nossa estrutura representam utilizadores, artigos, encomenda, transportadora.

Ao longo do decorrer deste projeto, verificamos que seria necessária uma classe que agregasse estas mesmas classes para que estas pudessem interagir entre si e, ainda, precisaríamos de uma classe capaz de interagir com o utilizador. Assim, decidimos proceder à implementação do modelo MVC não só para obter independência das várias camadas (Modelo, Vista e Controlador), mas também para ser possível interagir com o utilizador de uma forma segura.

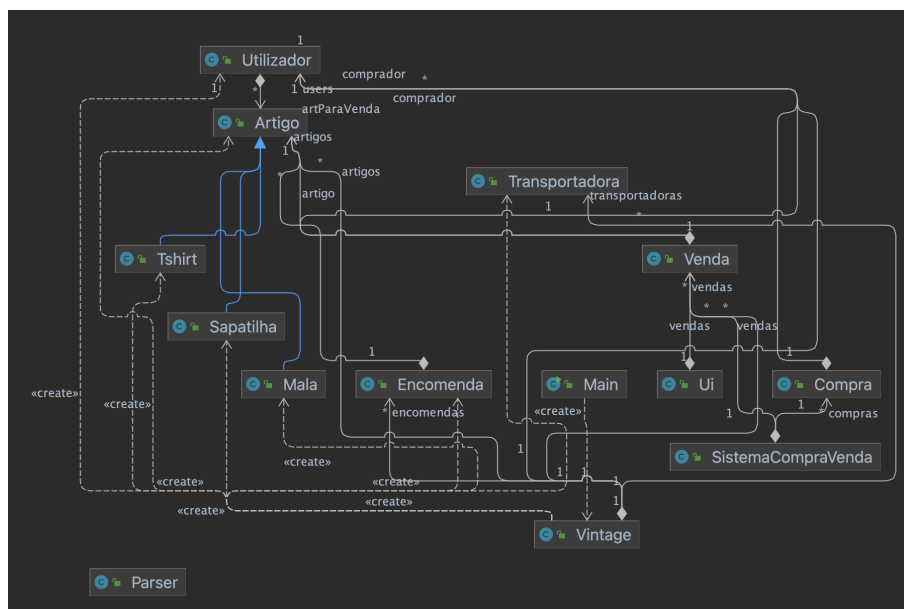


Figura 1: Diagrama de Classes

## 2.1 Main

Esta classe é a main do programa. É a classe que contém os vários módulos do MVC (Model-View-Controller) e que executa o interpretador.

## 2.2 Vintage

Esta é a classe responsável pelo modelo do programa. Aqui encontram-se todas as estruturas do nosso programa.

### 2.2.1 Atributos

- ArrayList<Utilizador>users - Lista de users;
- ArrayList<Transportadora>transportadoras - Lista de transportadoras;
- ArrayList<Encomenda>encomendas - Lista de encomendas;
- ArrayList<Artigo>artigos - Lista de artigos;
- int usersNum - Define o número do user;
- int transNum - Define o número da transportadora;
- int encNum - Define o número da encomenda;
- int artNum - Define o número do artigo;

## 2.3 Ui

Aqui é onde a interface é imprimida para a consola, contem apenas os prompts para guiar o utilizador.

## 2.4 Parser

O parser é responsável pela leitura do ficheiro, linha a linha, e chama as funções que analisam o conteúdo dos csv's.

## 2.5 Utilizador

Esta é a classe com a informação de um utilizador.

### 2.5.1 Atributos

- String codigo - Código dado pelo sistema para cada utilizador;
- String email - Email do utilizador para o login;
- String password - Password do utilizador;
- String nome - Nome do user;
- String morada - Morada do user;
- String numFiscal - N<sup>o</sup> fiscal do user;
- ArrayList<Artigo>artParaVenda - Lista de artigos para a venda com o seu preço;
- ArrayList<Artigo>artVendidos - Lista de artigos vendidos com o seu preço;
- ArrayList<Artigo>artComprados - Lista de artigos comprados com o seu preço;

## 2.6 Transportadora

Esta classe é responsável pelo tratamento de dados relacionados com a transportadora, quer seja o calculo do custo de as taxas de entrega, quer seja a guardar as informações de cada transportadora

### 2.6.1 Atributos

- String nome - Nome da transportadora;
- Double valorBasePequeno - Valor base da expedição para encomendas pequenas (1 artigo); médias (2 a 5 artigos) e grandes (mais que 5 artigos).
- Double valorBaseMedio - Valor base da expedição para encomendas médias (2 a 5 artigos);
- Double valorBaseGrande - Valor base da expedição para encomendas grandes (mais que 5 artigos);
- Double impostoMultiplicador - Valor do imposto a cobrar em cada transporte;
- Boolean isEspecializada - Verifica se é premium ou não;
- Boolean isEspecializadaPremium - Verifica se é premium ou não;

## 2.7 Encomenda

Esta classe trata de guardar os dados de uma encomenda.

### 2.7.1 Atributos

- String codVendedor - Código do vendedor que efetuou a encomenda;
- String codCliente - Código do cliente que recebe a encomenda;
- Int estado - Define o estado em que uma encomenda se encontra (0 - pendente da compra; 1 - entregue finalizado);
- ArrayList<Artigo>artigos - Lista de artigos que pertencem à encomenda;
- Int dimensao - Tamanho da encomenda;
- LocalDate dataenc - Data em que a encomenda foi criada;
- String nometransportadora - Nome da transportadora que vai efetuar a entrega da encomenda;

## 2.8 Artigo

Esta é a classe principal de um artigo, ou seja, os 3 artigos que vão ser apresentados a seguir, todos dão **"extend"** a este. Nesta classe, são guardados todos os dados comuns a todos os artigos.

### 2.8.1 Atributos

- String cod - Código do artigo;
- Boolean usado - Verifica se o artigo foi usado ou não(0 - usado; 1 - não usado);
- Double estado - Descreve o estado do artigo (1 - novo de fábrica; 0.8 - pouco uso; 0.6 - Aceitável; 0.4 - Muito uso; 0.2 - Muito desgastado);
- Int donos - Número total de donos;
- String descricao - Faz uma pequena descrição do produto;
- String marca - Marca do produto;
- Double preco - Preço base do artigo;
- Int desconto - Desconto em percentagem, que se aplica ao artigo;
- Boolean isPremium - Verifica se o artigo é **"premium"** ou não;

## 3 Mala

Esta classe refere-se aos artigos do tipo Mala, onde são guardadas as características específicas desta classe.

### 3.0.1 Atributos

- Int ano - Ano da coleção a que a mala faz parte;
- Int dimensao - Tamanho do artigo (Em litros - 2l, mais pequena; 60l - maior);
- Double material - Representa a qualidade do material (De 0 a 1);
- LocalDate lancamento - Data de lançamento da coleção a que pertence;

## 3.1 Sapatilha

Esta classe refere-se aos artigos do tipo Sapatilha, onde são guardadas as características específicas desta classe.

### 3.1.1 Atributos

- Int ano - Ano da coleção a que faz parte;
- Boolean atacadores - Verifica se tem atacadores ou não;
- String cor - Representa a cor da sapatilha;
- LocalDate lancamento - Data de lançamento da coleção a que pertence;

## 3.2 T-shirt

Esta classe refere-se aos artigos do tipo T-shirt, onde são guardadas as características específicas desta classe.

### 3.2.1 Atributos

- String tamanho - Tamanho da T-shirt (S, M, L, XL);
- String padrao - Padrão desenhado na T-shirt (LISA, RISCAS, PALMEIRAS);

## 4 Descrição da Aplicação

O modo de interação com o utilizador escolhido foi uma interface de linha de comandos.

### 4.1 Menu Principal

Quando o cliente entra no programa depara-se com o menu de boas vindas. Daí, este pode premir qualquer tecla para exibir o menu principal.

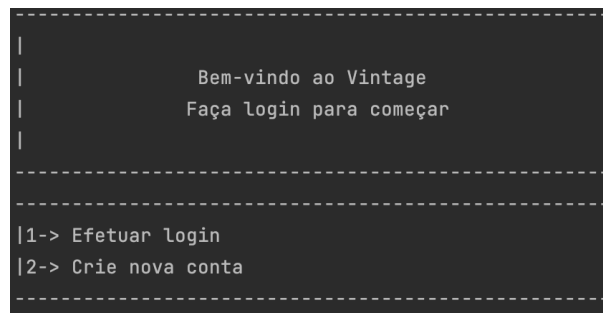


Figura 2: Menu Principal

Apresenta as opções de efetuar o login ou criar umz conta nova.

### 4.2 Menu Utilizador

Após o login ser efetuado, aparece um novo menu com as seguintes opções:

- 1 - Adicionar produto para venda
- 2 - Navegar produtos à venda (meus produtos)
- 3 - Navegar produtos à venda
- 4 - Efetuar encomenda
- 5 - Verificar estado das minhas encomendas
- 6 - Guardar estado do programa



```
-----
|1-> Adicionar produto para venda          |
|2-> Navegar produtos à venda (meus produtos)|
|3-> Navegar produtos à venda             |
|4-> Efetuar encomenda                     |
|5-> Verificar estado das minhas encomendas |
|6-> Guardar estado do programa            |
|q-> Sair                                  |
-----
```

Figura 3: Menu Utilizador

Ao decidir adicionar um produto para venda, o utilizador tem a opção de definir como se caracteriza este mesmo produto (se é novo ou não), o estado do mesmo, o número de donos que o produto já apresentou e uma breve descrição sobre o mesmo, bem como caracterizar os seus atributos.

O user tem também a opção de ver a lista dos seus produtos que se encontram à venda.

```
-----
U000001
M000000;Novo;;Donos:
1;malã de couro;Marca: guess;Preco: 180.0€;Ano colecao: 2015;Dimensao: 20;Material: 1.0;Lancamento: 2015-06-22

S000000;Novo;;Donos:
2;novo;Marca: adidas;Preco: 60.0€;Tamanho: 37;Tem Atacadores;Cor: branco;Lancamento: 2010-06-28

T000000;Novo;;Donos:
3;usada poucas vezes;Marca: levis;Preco: 20.0€;Tamanho: S;Padrao: liso
-----
```

Figura 4: Lista dos produtos à venda de um certo user

O user tem também a opção de ver a lista dos produtos que se encontram à venda.

```
U000001
T000000011;Novo;;Donos:
0;Camisola desportiva;Marca: Nike;Premium;Preco: 30.0€;Tamanho: M;Padrao: RISCAS

S000019;Novo;;Donos:
1;louco;Marca: spipe;Preco: 0.0€;Tamanho: 40;Não Tem Atacadores;Cor: preto;Lancamento: 2023-05-03

-----
U000002
-----
U000003
T000000013;Novo;;Donos:
4;Com algum uso;Marca: C&A;Preco: 19.0€;Tamanho: S;Padrao: RISCAS
```

Figura 5: Lista dos produtos à venda

Pode, depois, efetuar uma encomenda, tenho de escolher o código do user que está a vender o produto pretendido, bem como a transportadora desejada.

```
-----
4
-----
|          Insira o código do Vendedor          |
| l-> listar Vendedores e seus produtos          |
|-----|
U000003
-----
U000003
T000000013;Novo;;Donos:
4;Com algum uso;Marca: C&A;Preco: 19.0€;Tamanho: S;Padrao: RISCAS
-----
|          Insira o código dos produtos          |
| Separados por ';' e sem espaços                |
|-----|
T000000013
-----
|Escolha a transportadora que deseja efetuar encomenda|
-----
Transportadora{nhl', valorBasePequeno=1.0, valorBaseMedio=1.5, valorBaseGrande=1.8, impostoMultiplicador=12.0}
Transportadora{ukz', valorBasePequeno=0.7, valorBaseMedio=1.8, valorBaseGrande=2.1, impostoMultiplicador=10.0}
Transportadora{zpp', valorBasePequeno=1.7, valorBaseMedio=1.7, valorBaseGrande=1.7, impostoMultiplicador=115.0}
Transportadora{eg', valorBasePequeno=0.6, valorBaseMedio=0.9, valorBaseGrande=1.2, impostoMultiplicador=12.0}
ukz
Encomenda:Estado da Encomenda: 0;Artigos da encomenda: [T000000013;Novo;;Donos:
4;Com algum uso;Marca: C&A;Preco: 19.0€;Tamanho: S;Padrao: RISCAS
];Dimensão de encomenda: 1;Data da encomenda: 2023-05-14;Nome da transportadora: ukz;Preço final: 19.0
```

Figura 6: Efetuar uma encomenda

Pode, também, ter acesso às estatísticas do programa, como quem foi o vendedor e transportadora que mais faturaram, as encomendas emitidas por um utilizador e por fim o lucro total da aplicação Vintage.

```
9
-----
|          Insira o código do Vendedor          |
|-----|
U000007
-----
Encomendas: [Estado da Encomenda: 1;Artigos da encomenda: [];Dimensão de encomenda: 1;Data da encomenda: 2023-04-03;Nome da transportadora: nhl
```

Figura 7: Encomendas emitidas por um user

```
8
-----
| Foi o vendedor: Cristiano Ronaldo dos Santos AveiroU000007 |
|-----|
```

Figura 8: Vendedor que mais faturou

```
7
-----
| Foi a transportadora: zpp |
|-----|
```

Figura 9: Transportadora que mais faturou

```
6
-----
| Lucros totais =
0.0
|
-----
```

Figura 10: Lucro final

## 5 Conclusão

Para finalizar, consideramos que a nossa solução atingiu os objetivos do projeto com sucesso e estamos satisfeitos com seu desempenho. Embora tenhamos enfrentado alguns inconvenientes, acreditamos que se a aplicação fosse executada em tempo real, seria muito prática. Durante a realização do projeto, aprendemos a importância da segurança do código e da organização das estruturas de dados, bem como a influência significativa que diferentes algoritmos e relações entre eles tiveram nas funcionalidades do programa. De um modo geral, consideramos o projeto um sucesso e estamos satisfeitos com as lições aprendidas.