



Licenciatura em Engenharia Informática

# Programação Orientada a Objetos Relatório Final

GRUPO 85

António Pedro (a100821) João Rodrigues (a100896) Mateus Lemos(a100645)  
Ano Letivo de 2022/2023

---

## Índice

|                      |   |
|----------------------|---|
| Introdução .....     | 1 |
| Arquitetura .....    | 1 |
| Funcionalidades..... | 4 |
| Conclusões .....     | 5 |

---

## Introdução

Com o aproximar do fim do ano letivo, de igual forma, o projeto realizado no âmbito da Unidade Curricular de Programação Orientada a Objetos é concluído. Terá como objetivo, por conseguinte, o presente relatório para ajudar quem o leia a melhor compreender e interpretar o trabalho desenvolvido pelo grupo, bem como as metodologias empregues, dando ainda espaço a uma reflexão sobre o mesmo.

Tendo como propósito o desenvolvimento de um sistema de *marketplace* denominado *Vintage*, que permitisse a compra e venda de artigos novos e usados, por parte de utilizadores, que poderiam ser simultaneamente compradores e vendedores dentro do dito *marketplace*, na linguagem de *Java*, este projeto exigia a criação de diversas classes para descrever e armazenar a informação relativa tanto a esses artigos e utilizadores. Em adição, seria necessário também a inclusão de classes para as transportadoras das encomendas, gerir todas as compras e vendas dos artigos, incluindo o seu stock e o transporte efetuado pelas referidas transportadoras, e efetuar cálculos estatísticos sobre essas mesmas operações.

Deste modo, e com a finalidade de favorecer a rápida elaboração e conclusão deste trabalho, foi projetada logo de início toda a arquitetura e estrutura do projeto. Durante tal processo, foi também evidenciada aquela que corretamente se previu como a maior dificuldade: o prazo temporal reduzido, tendo em conta, não a dificuldade, mas a dimensão do projeto, e o período de execução coincidir no seu período intermédio, mas principalmente final com uma alta densidade de outros trabalhos e avaliações.

Tendo em conta essa limitação, começamos por decidir uma abordagem mais simplista ao projeto que permitisse a elaboração e conclusão de certos objetivos mais fáceis de atingir em termos de tempo empregue, e que permitisse ir adicionando funcionalidades ao sistema caso se revelasse possível em termos temporais.

## Arquitetura

Como mencionado anteriormente, devido à abordagem faseada que decidimos tomar optámos pela não utilização do modelo VMC, tendo o caminho seguido permitido maior liberdade e fluidez na criação de classes, funções, e consequentemente, no acréscimo de funcionalidades.

Assim, começamos por definir a classe *main*, que dado os argumentos recebidos se limita a invocar a classe *Vintage*. Esta última é o coração do nosso projeto, sendo responsável pela interação e coordenação entre as restantes classes, pela interpretação de comandos do utilizador, parsing de ficheiros de entrada, e apresentação dos resultados, em suma, gere todas as outras componentes de forma a assegurar o correto funcionamento do sistema. Seguidamente foram criadas as seguintes classes: *Tshirt*, *Sapatilha*, *Mala*, *Artigo*, *Transportadora*, *Encomenda*, e *User*, tendo, para as funcionalidades básicas, sido definidos, os seus atributos, e as funções, de criação das classes em diversos cenários, *clone*, *toString*, *equals*, bem como os diversos *setters* e *getters*.

Imediatamente foram criadas as classes *CreateTshirt*, *CreateSapatilha*, *CreateMala*, *CreateTransportadora*, *CreateEncomenda*, e *CreateUser*, que dada uma *string* contendo dados para a criação da classe referida no nome da função, cria uma instância dessa classe com esses dados.

---

Mais tarde, foram criadas as classes *ParseArtigo*, *ParseTransportadora*, *ParseEncomenda*, e *ParseUser*, que permitem a criação de uma ou várias instâncias das referidas classes através de um ficheiro de texto, e ainda a classe *ParseData*, que permite receber e alterar a data do sistema a partir de um ficheiro de texto.

Foram ainda criadas as classes *InterativoAdd*, *InterativoChange*, e *InterativoStats*, para suportar a interface com o utilizador.

As classes *Artigo*, *CreateEncomenda*, *Mala*, *ParseUser*, *Sapatilha*, *Transportadora*, *Tshirt*, *Users*, e *Vintage*, foram atualizadas para suportar os processos estatísticos requeridos. Tal, causou a necessidade de que também a classe *InterativoStats*, que gere a forma como as estatísticas são apresentadas ao utilizador, fosse alterada.

Finalmente foram corrigidos alguns erros relacionados com a alteração, finalização e guardar de encomendas que viram as classes *Vintage* e *Encomenda* serem modificadas. Infelizmente por motivos de limitação temporal, encerrámos neste ponto o desenvolvimento do nosso programa, ou seja com as funcionalidades de gestão das entidades e cálculo de estatísticas sobre o estado de programa. O programa prevê também a existência de produtos *Premium*, mas as transportadoras *Premium* apesar de iniciadas, não se encontram funcionais, pelo que não serão incluídas na entrega do trabalho.

De seguida apresentaremos o diagrama de classes deste projeto (*Figura 1*).



---

## Funcionalidades

No momento de entrega deste projeto, o sistema é capaz da total gestão das entidades, sendo possível adicionar e remover *Users*, compradores e vendedores, adicionar e remover *Artigos*, dos diversos tipos, *Tshirts*, *Malas*, *Sapatilhas*, que possuem todos os atributos básicos requeridos e ainda a possibilidade de serem considerados *Premium*, bem como criar, guardar e finalizar encomendas, suportando ainda receber informações via linha de comando no terminal, ou seja o utilizador especificar no momento todas as características daquilo que deseja criar ou adicionar via interface, ou introduzir na mesma o caminho para um ficheiro de texto previamente preparado com os diversos elementos que deseja adicionar ao sistema.

A nível de estatísticas o programa é capaz de determinar certos valores requeridos, como por exemplo a faturação de users e transportadoras, assim como o dinheiro ganho pela própria vintage, ou até mesmo os artigos vendidos por um user específico. (Demonstrar algum funcionamento do programa, po-lo a correr executar pelo menos a adição de algo, a criação de uma encomenda, a alteração da data do sistema, e dois cálculos estatísticos a escolha).

Temos de seguida exemplos de funcionamento de alguns comandos do programa:

```
Add Encomenda Encomenda;CodComprador;estado;data;nrartigos;artigo1;artigoN
Add User User;codigo;email;nome;morada;nif
Add Transportadora Transportadora;nome;basePequena;baseMedia;baseGrande;imposto;lucro;tempoexpedicao
Exit
Save

Add Encomenda Encomenda;749205;pendente;2020-04-16;3;h40g1;1efr3;r1j8d
exit
```

Figura 2 – Adição de encomenda pelo utilizador pela interface terminal

```
Encomenda;749205;Fedex;pequena;26.391;pendente;2020-04-16;1;r1j8d
Encomenda;749205;CTT;media;316.506;pendente;2020-04-16;2;h40g1;1efr3
```

Figura 3 - Verificação que a encomenda foi adicionada corretamente, tendo sido dividida em duas encomendas devido às diferentes transportadoras.

---

```
UsersMaiorFaturacaoOrdenados 2018-01-01 2023-01-01
```

```
150347;Guilherme;237.9999999999997  
123123;Josefino;77.0  
865397;Alberta;25.0  
749205;Luís;10.0  
937501;Antonietta;0.0  
527501;Daniel;0.0
```

*Figura 4* - Verificação estatística. Lista ordenada dos utilizadores que mais faturaram num dado período de tempo.

## Conclusões

Com o encerramento deste projeto, podemos afirmar que a nossa destreza na linguagem *Java*, se consolidou bastante, tendo em conta que se tratou do nosso primeiro contacto com esta linguagem, destacam-se a fácil criação e manipulação de objetos, já característicos da linguagem, que nos agradaram bastante. Apesar de estarmos moderadamente satisfeitos com o que alcançamos no projeto, tendo em conta o tempo que tivemos disponível para a execução do mesmo, e com aquilo que aprendemos resultante de tal, permanece o sentimento de que podíamos ter feito mais, dado mais tempo, e que bastaria mais uma semana para concluir todas as funcionalidades dos vários níveis presentes no enunciado, e obter uma excelente aprendizagem da linguagem e avaliação na Unidade Curricular.

Por fim, consideramos que não alcançamos todos os objetivos propostos, mas um pouco mais que os esperados inicialmente, tendo o planeamento inicial sido essencial para tal. Por isso, o balanço que fazemos do nosso aproveitamento neste projeto é algo positivo.