



Universidade do Minho
Escola de Engenharia

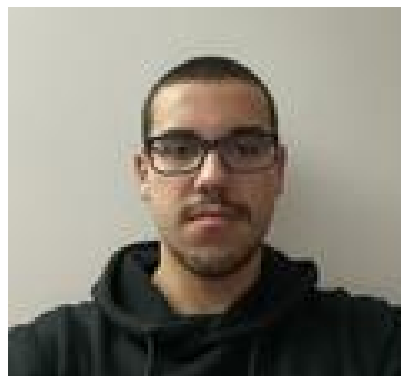
Relatório do Projeto prático de Programação Orientada aos Objetos

Grupo 56

Samuel Macieira Ferreira
(a100654)

Pedro Miguel Costa Azevedo
(a100557)

Miguel Tomás Antunes Pinto
(a100815)



Introdução

Este projeto em Java apresenta a elaboração de um sistema de marketplace Vintage que permite a compra e venda de artigos novos e usados (Sapatilhas, Malas, T-Shirt). O sistema foi projetado para permitir que os utilizadores atuem como vendedores e compradores e façam as suas compras através da plataforma.

Este relatório apresenta uma visão geral da concepção do sistema, descrevendo suas funcionalidades, características e requisitos. Espera-se que este relatório sirva como base para o desenvolvimento e implementação do sistema de marketplace vintage.

Main

Como seria de esperar, a execução começa pela nossa Classe Main, que inicia a execução do Controller.

Controller

Esta classe é responsável por gerir o fluxo da aplicação. Interage principalmente com a classe Menu, a fim de dispor todos os menus necessários à interação com o utilizador pelo terminal.

Menu

O Menu é a Vista do programa. Interage com o Controller, com o objetivo de enviar ou receber (input/output) informação pelo terminal, bem como mostrar mensagens de erro. Possui todos os *prints* necessários para conceder a estética do programa e trazer uma interação agradável e eficaz com o utilizador através de todos os menus.

Parse

A esta classe cabe o trabalho de fazer o parsing de dados, quando é pedido que se carregue um estado a partir de um ficheiro de texto, utilizando o ficheiro “ficheiro.txt” criado por nós ao longo do Projeto. Possui o método *parsing*, que parte o ficheiro linha a linha e trata de criar os objetos necessários e/ou adicioná-los no local requerido.

Sys

```
private Map<String, Utilizador> utilizador;  
private Map<String, Transportadora> transportadora;  
private Map<String, Produto> produtos;  
private List<Encomenda> encomendas;  
private List<Fatura> faturas;  
private int id;
```

A classe Sys é o sistema do nosso programa com a finalidade de gerir toda a informação, possuindo três Map's que correspondem a todos os Utilizadores, Transportadoras e Produtos, duas listas uma com todas as Faturas e outra com todas as Encomendas e ainda um id que serve para incrementar o id dos Produtos e Utilizadores. Trata-se muito provavelmente da classe fundamental a toda a gestão de dados.

Utilizador

```
private String id;  
private String email;  
private String nome;  
private String morada;  
private String numFiscal;  
private List<Produto> produtosAVenda;  
private List<Produto> produtosVendidos;  
private List<Produto> produtosAdquiridos;
```

A classe Utilizador representa um indivíduo que tanto pode atuar como comprador ou vendedor, possui um id, um email, um nome, uma morada, um número Fiscal e ainda três Listas, uma para Produtos à Venda, uma para produtos Vendidos e outra para Produtos Comprados.

Produto

```
private String descricao;  
private String marca;  
private String codigo;  
private Estado estado;  
private double precoBase;  
private double correcaoPreco;  
private String proprietario;
```

A classe Produto é a superclasse de todas as outras classes que representam Produtos (T-Shirt, Sapatilha, Mala), sendo a descrição, a marca, o código (id), o estado, o preço Base, a correção de preço e o Proprietário do Produto os atributos comuns a todas essas entidades.

T-Shirt

```
private String tamanho;  
private Padrao padrao;
```

Trata-se de uma subclasse de Produto e como tal, além de possuir os atributos de Produto possui também variáveis de instâncias específicas a si como o tamanho e o padrão.

Mala

```
private double dimensao;  
private String material;  
private int anoColecao;  
private boolean premium;
```

Trata-se de uma subclasse de Produto e como tal, além de possuir os atributos de Produto possui também variáveis de instâncias específicas a si como a dimensão, o material, o ano de Coleção e se este é um produto Premium.

Sapatilha

```
private int tamanho;  
private boolean possuiAtacadores;  
private String cor;  
private int anoColecao;  
private boolean premium;  
private boolean isUsado;  
private int numeroDonos;  
private double estadoUtilizacao;
```

Trata-se de uma subclasse de Produto e como tal, além de possuir os atributos de Produto possui também variáveis de instâncias específicas a si como o tamanho, se possui atacadores, a cor, o ano da coleção, se é um produto premium, se já foi usado, o número de donos e o estado de Utilização da Sapatilha

Encomenda

```
private List<Produto> produtos;  
private String nomeVendedor;  
private String nomeComprador;  
private String nomeTransportadora;  
private Embalagem embalagem;  
private EstadoEncomenda estadoEncomenda;  
private double taxaNovo;  
private double taxaUsado;  
private double custoExpedicao;  
private DateFormat dataCriacao;  
private DateFormat dataExpedicao;
```

A classe Encomenda é onde acontece a junção de todas as outras classes, possui uma Lista de Produtos, o nome do Vendedor, o nome do Comprador, o nome da Transportadora, o tamanho da Embalagem, o estado da Encomenda, a taxa de Novo e a taxa de Usado, o custo de Expedição, a data de Criação da Encomenda e a data de Expedição.

Fatura

```
private String nomeVendedor;  
private String nomeComprador;  
private String nomeTransportadora;  
private List<Produto> produtos;  
private double precoTotal;
```

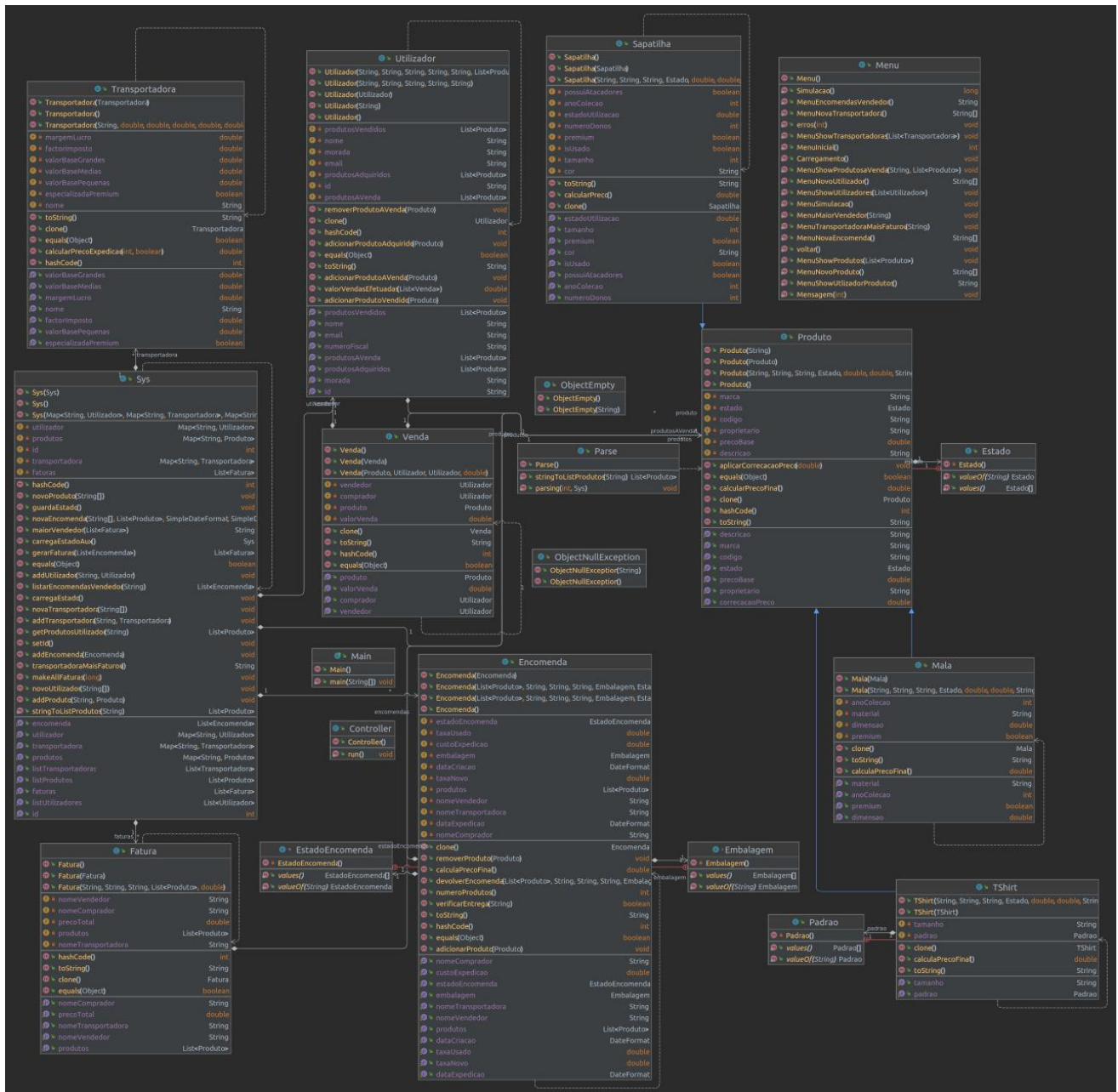
Esta classe identifica uma fatura, que é gerada a qualquer momento de execução do programa, e que, aquando de uma simulação, guarda as informações necessárias para gerar a própria.

Transportadora

```
private String nome;  
private double valorBasePequenas;  
private double valorBaseMedias;  
private double valorBaseGrandes;  
private double factorImposto;  
private double margemLucro;  
private boolean especializadaPremium;
```

A classe Transportadora é a que está encarregue de entregar a encomenda ao Utilizador, possui um nome, um valor para Encomendas Pequenas, Médias ou Grandes, um Imposto, uma margem de lucro e ainda se está especializada em Produtos Premium.

Diagrama de Classes



Descrição da Aplicação Desenvolvida

O nosso programa inicia com a disposição do seguinte menu

```
#####  
#                                     #  
# ----- VINTAGE ----- #  
# (1) Carregar ficheiro          #  
# (2) Salvar ficheiro            #  
# (3) Verificar Utilizadores existentes #  
# (4) Verificar Transportadoras  #  
# (5) Verificar Lista de Produtos #  
# (6) Verificar Produtos á Venda #  
# (7) Adicionar um Utilizador    #  
# (8) Adicionar um Produto       #  
# (9) Adicionar uma Transportadora #  
# (10) Fazer um Encomenda        #  
# (11) Vendedor que mais faturou  #  
# (12) Transportadora que mais faturou #  
# (13) Encomendas de um vendedor  #  
# (14) Lista maiores compradores  #  
# (15) Lista maiores vendedores   #  
# (16) Simulação (Avançar no Tempo) #  
# (17) Faruração da VINTAGE       #  
#                                 #  
# (0) Sair                        #  
#                                 #  
#####  
  
Selecione uma das opções acima:
```

Este menu é invocado de volta sempre que é selecionada uma opção inexistente, ou depois de ser efetuada alguma operação instantânea. É-nos possível navegar por (quase) todas as opções sem que o programa termine, evitando ter de o iniciar novamente para concluir novos pedidos.

(1) Carregar Ficheiro

Esta opção permite-nos carregar um estado de um ficheiro de texto (Carregando do ficheiro “ficheiro.txt” ou de um ficheiro em binário)

(2) Salvar Ficheiro

Esta opção salva o nosso estado de execução atual do programa.

(3) Verificar Utilizadores Existentes

Imprime no terminal todos os Utilizadores existentes, bem como os seus atributos.

(4) Verificar Transportadoras

Imprime no terminal todas as Transportadoras existentes, bem como os seus atributos

(5) Verificar Lista de Produtos

Imprime no terminal todos os Produtos existentes, bem como os seus atributos

(6) Verificar Produtos à Venda

Pede-nos para inserir o nome de um Utilizador e mostra-nos quais os produtos que este tem à venda.

(7) Adicionar um Utilizador

Permite-nos adicionar um novo utilizador, inserindo os seus atributos

(8) Adicionar um Produto

Permite-nos adicionar um novo produto, inserindo os seus atributos

(9) Adicionar uma Transportadora

Permite-nos adicionar um nova Transportadora, inserindo os seus atributos

(10) Fazer uma Encomenda

Permite-nos fazer uma nova encomenda

(11) Vendedor que mais Faturou

Diz-nos até á data qual o Utilizador que mais faturou

(12) Vendedor que mais Faturou

Diz-nos até á data qual a Transportadora que mais faturou

(13) Encomendas de um Vendedor

Pede-nos que seja fornecido um nome e então lista todas as encomendas desse Utilizador

(14) Lista de maiores Compradores

Imprime no terminal os 5 maiores compradores

(15) Lista de maiores Vendedores

Imprime no terminal os 5 maiores Vendedores

(16) Simulação (Avançar no Tempo)

Esta opção serve para iniciar as simulações, indicamos a data até à qual pretendemos simular e recebemos a quantidade de dias que estamos a simular. Neste momento são criadas as faturas.

(17) Faturação da VINTAGE

Esta opção diz-nos qual a faturação da Vintage depois de feitas e contabilizadas todas as encomendas

Conclusões

Todo o processo de desenvolvimento deste trabalho prático, contribuiu fortemente para a nossa consolidação de matéria lecionada nas aulas teóricas. Refletindo acerca do nosso produto final, pensamos que poderíamos ter tido um melhor aproveitamento no projeto mesmo assim, acabamos por cumprir com parte dos objetivos delineados no enunciado e fruto do nosso esforço concluímos este trabalho. Acreditamos que respeitamos os conceitos de modularidade e encapsulamento fundamentais e fizemos frente às maiores dificuldades em todo o percurso.