

Projecto prático de Programação Orientada aos Objectos

LEI + LCC

edição 22/23 - v2.0

Conteúdo

1	Introdução	1
1.1	Os <i>Artigos</i>	2
1.2	Os <i>Utilizadores</i>	2
1.3	As <i>Encomendas</i>	3
1.4	As <i>Transportadoras</i>	3
2	O Funcionamento do programa	3
3	Requisitos	4
3.1	Requisitos base de gestão das entidades	4
3.2	Efectuar estatísticas sobre o estado do programa	4
3.3	Alterar os transportadores e prever a noção de <i>Premium</i>	4
3.4	Automatizar a simulação	5
4	Relatório	5
5	Salvaguarda do estado da aplicação	5
6	Cronograma	6

1 Introdução

Considere que se pretende construir um sistema de *marketplace* **Vintage** que permite a compra e venda de artigos novos e usados de vários tipos. No *marketplace*, os utilizadores registados assumem o papel de vendedor ou comprador. Os vendedores podem adicionar novos artigos para venda. Os compradores podem optar pela compra dos artigos disponíveis. Todas as compras e vendas são geridas a partir deste sistema.

Os artigos podem ser de vários tipos, sendo em particular artigos de vestuário, calçado, acessórios, mas não só. Espera-se que na fase de concepção cada grupo tenha em linha de conta que o *marketplace* possa futuramente ser expandido a outros tipos de produto.

Cada artigo é caracterizado por um conjunto comum de propriedades, sendo que decorrem depois especializações dependendo do tipo de artigo em questão. Cada artigo é identificado por um código alfanumérico (i.e., código de barras) único.

A compra dos artigos é organizada em encomendas que são posteriormente expedidas por várias empresas de transporte. Por cada encomenda finalizada, a **Vintage** cobra uma taxa de garantia de serviço ao vendedor, garantindo a segurança e satisfação dos compradores caso as encomendas não sejam entregues.

O sistema **Vintage** mantém controlo sob o stock dos artigos disponíveis, bem como das encomendas efetuadas pelos compradores e das vendas efetuadas pelos vendedores.

Deverá ser possível a um vendedor publicar os seus artigos, bem como a um comprador optar por vários artigos e encomendá-los. No momento em que o vendedor publica os seus artigos para venda, ele decide também qual é a empresa transportadora que tratará da expedição.

Deverá ainda ser possível ao comprador adicionar vários artigos a uma encomenda e concluir a compra. Deverá também ser-lhe possível devolver a encomenda, mas apenas 48 horas (que poderá ser configurável) após a entrega da encomenda.

Um aspecto que é importante ter em linha de conta é que, para que o sistema possa ser simulado o mais possível num cenário de utilização real, será necessário efectuar *saltos no tempo*. Isto é, o sistema deverá saber em que data é que se encontra e deverá estar explícita uma funcionalidade que permitirá avançar para uma data no futuro. O avançar do tempo (simulado por questões práticas e de redução de complexidade) deverá originar todos os movimentos que estarão pendentes, ou seja, as encomendas são entregues, o stock de cada produto é reduzido, etc.

Nota: esta versão tem alterações de pormenor, em relação à primeira versão, no sentido de detalhar melhor os estados pela qual passa a encomenda.

1.1 Os Artigos

A *Vintage* especializa-se em três tipos principais de artigos: sapatilhas, t-shirts e malas. Cada um destes artigos pode ser novo ou usado, sendo neste último caso disponibilizado também uma avaliação do seu estado e informação sobre o número de donos que já possuíram o artigo. Cada artigo contém uma descrição, uma marca, um código alfanumérico e um preço base e ainda uma correcção de preço (normalmente um desconto) que é definida em função de cada tipo de artigo e da sua condição particular.

Os artigos são detalhados tal que:

- As *Sapatilhas*, possuem um tamanho numérico, uma indicação se possuem atacadores/atilhos, uma cor e a data de lançamento da coleção a que pertencem (existe uma coleção por ano). Apenas há lugar a aplicação de desconto em sapatilhas usadas (ou seja que não sejam desta coleção/ano), definido pelo vendedor, ou em sapatilhas novas acima do tamanho 45. Um exemplo de cálculo do preço de uma sapatilha usada, ou antiga, poderá ser algo como

$$precoBase - (precoBase/numeroDonos * estadoUtilizacao)$$

Poderá ainda existir um tipo especial de *Sapatilhas*, as *Premium* de edições especiais de autores reconhecidos, cujo valor de mercado aumenta com o passar dos anos. Para essas a fórmula de cálculo do valor terá de prever não um desconto, mas um acréscimo de valor.

- As *T-Shirt*, possuem um tamanho (S,M,L,XL) e um padrão (liso, riscas, palmeiras). As T-Shirts com padrão liso nunca têm desconto. Os restantes padrões têm um desconto fixo de 50% se forem usados.
- As *Malas*, possuem informação sobre a sua dimensão, sendo que o desconto será sempre proporcionalmente inverso à dimensão, o material de que são constituídas e o ano da coleção. Tal como para as *Sapatilhas*, para as *Malas* existem aquelas que são *Premium* e cujo valor em vez de decrescer com a utilização (número de anos da carteira) aumenta com a mesma. Essas malas apresentarão uma valorização de $X\%$ ao ano (a definir por cada tipo de mala).

1.2 Os Utilizadores

Para cada utilizador deverão ser guardadas as informações relativas ao seu código no sistema (fornecido automaticamente pelo sistema), ao seu email (que deverá ser o login), nome, morada, número fiscal e deverá ser guardada toda a informação sobre produtos que tem à venda ou que já vendeu, bem como os que já adquiriu. É suposto o sistema guardar a informação sobre as vendas que foram efectuadas e o valor que delas resultou. Note que um mesmo utilizador poderá actuar como vendedor e como comprador.

1.3 As *Encomendas*

Cada encomenda possui uma colecção (no sentido lato do termo) de *Artigos* nela contidos, a dimensão da embalagem para expedição (grande, médio e pequeno), o preço final aplicados todos os cálculos, com a taxa de satisfação de serviço de 0,5 € por cada artigo novo e de 0,25€ por cada artigo usado e ainda os custos de expedição (que serão dependentes de transportadora para transportadora). Possui ainda um estado. A encomenda é iniciada no estado pendente, passando ao estado finalizada quando a compra é concluída. Finalmente passa ao estado expedida quando a transportadora a tiver enviado. A encomenda possui ainda a data em que foi criada.

Deverá ser possível:

- Criar uma encomenda.
- Adicionar e remover artigos de uma encomenda.
- Calcular o preço final a pagar pela encomenda.

, bem assim como devolver a encomenda, caso esteja ainda dentro do prazo limite para o fazer.

1.4 As *Transportadoras*

Existem no mercado diversas transportadoras e cada uma delas tem uma definição de preço que cobra pela expedição de acordo com os seus critérios comerciais. Cada artigo terá uma transportadora atribuída que é escolhida pelo vendedor no momento de publicação do artigo para venda, não podendo haver artigos para venda sem uma transportadora atribuída.

As diversas transportadoras têm liberdade na construção da fórmula que determina o valor da expedição, mas devem respeitar algumas premissas, a saber:

- Está definido no arranque do sistema qual é o valor base da expedição para encomendas pequenas (1 artigo), médias (2 a 5 artigos) e grandes (mais que 5 artigos).
- Está definido no arranque do sistema qual é o factor multiplicativo dos impostos, isto é, o sistema tem definido o valor do imposto a cobrar em cada transporte.

Fora estas premissas, as transportadoras podem criar a fórmula de cálculo do preço. Um exemplo seria algo como $PrecoExpedicao = (ValorBase * margemlucrotransportadora * (1 + Imposto)) * 0.9$.

Um factor de valorização do sistema será a existência de transportadoras especializadas na entrega de artigos *Premium*, sendo que nem todas as transportadoras existentes no sistema cumprem este requisito. O preço de expedição para um artigo *Premium* pode seguir uma regra de cálculo diferente dos outros artigos, sendo que essa decisão cabe apenas à transportadora.

2 O Funcionamento do programa

Pretende-se que com este programa se possa criar uma espécie de simulação da compra e venda de artigos na *Vintage*. Como foi atrás referido, terá de ser previsto que se possam efectuar mudanças de data (o que atrás se designou por *saltos no tempo*) e deverá também ser necessário entrar e sair do programa para simular acessos diferentes por parte de vendedores ou compradores. Mas isto ficará ao critério de cada grupo e da complexidade que pretender desenvolver.

1. o programa deve ter uma fase em que é permitido criar compradores/vendedores, artigos, e transportadores. Uma hipótese é esta informação ser carregada de um ficheiro, dispensando quem utiliza o programa de cada vez que o usar estar a fazer o setup da informação.
2. a componente de simulação deve começar já com uma configuração em que existem utilizadores, artigos, transportadores e encomendas.

3. o programa deve pedir ao utilizador que determine o avançar do tempo. Este avançar do tempo faz-se em dias e o programa deve permitir **ir para uma determinada data**. Nessa altura despacham-se as encomendas e fazem-se os cálculos dos totais dispendidos que são creditados ao vendedor e ao transportador e é actualizada a informação do sistema (diminui-se o stock dos produtos, etc.)
4. Quando se vai para uma determinada data e se despacham encomendas faz-se a emissão para cada vendedor/comprador da factura respectiva indicando a encomenda e o custo.
5. Os transportadores podem também alterar os valores de cálculo, mas mais uma vez isso só produz efeito nas próximas encomendas.

3 Requisitos

O trabalho proposto tem vários níveis de requisitos, desde os mais básicos até aos mais complexos. Para que os vários níveis de requisitos identificados sejam considerados cumpridos o programa deve respeitar os princípios da programação por objectos, ser robusto no tratamento de erros, bem como gravar, e também carregar, o estado do programa para um ficheiro de objectos. Deverá possuir uma camada de interacção com o utilizador, que se aconselha ser feita por menús em modo texto (atenção que não é solicitada a existência de camada gráfica), e que permita aceder às funcionalidades do programa. Durante a sessão de apresentação será também avaliada a forma como o programa está construído, nomeadamente a utilização das estruturas de dados e a forma como são utilizadas e manipuladas.

3.1 Requisitos base de gestão das entidades

Como requisitos base pretende-se ter a capacidade de criar utilizadores (vendedores/compradores), artigos, transportadores e fazer encomendas.

Nota máxima: 12 valores

3.2 Efectuar estatísticas sobre o estado do programa

Além do anterior, o programa deverá ser capaz de avançar o tempo, actualizar a informação do sistema e gerar as respectivas facturas. E ainda permitir saber:

1. qual é o vendedor que mais facturou num período ou desde sempre
2. qual o transportador com maior volume de facturação
3. listar as encomendas emitidas por um vendedor
4. fornecer uma ordenação dos maiores compradores/vendedores do sistema durante um período a determinar
5. determinar quanto dinheiro ganhou o **Vintage** no seu funcionamento

Nota máxima: 16 valores

3.3 Alterar os transportadores e prever a noção de *Premium*

Além dos requisitos anteriores, o programa deverá permitir satisfazer os requisitos identificados em 2.5, permitindo a alteração das transportadores e da sua fórmula de cálculo, bem como a existência de produtos e transportadoras *Premium*.

Nota máxima: 18 valores

3.4 Automatizar a simulação

A simulação sugerida até agora é feita por intervenção do utilizador sempre que este decide mudar explicitamente a data (para uma data no futuro). O sistema poderia, no entanto, admitir que se pudessem colocar eventos mais detalhados e possibilitando inclusive a compra e venda de artigos e a construção de encomendas, etc. Para tal deveria ler, por exemplo um ficheiro, com informação relativa *data, utilizador, artigo, accao, valor* e executar esses comandos.

Por exemplo:

```
2023.04.10, user1, air jordan, comprar, 500
2023.04.10, user1, tshirt Linux, comprar, 25
2023.04.11, transpostador1, 20 --> altera o valor base a comprar por qualquer encomenda
2023.04.12, user2, mala, alteraValorValorizacao, novoValor
```

Este ficheiro pode ser construído consoante cada grupo e pode até incluir além do dia informação sobre a hora em que ocorre a mudança. Estes requisitos representam um grau de complexidade acrescido e destinam-se a distinguir os trabalhos que sejam excelentes.

Nota máxima: 20 valores

Obviamente que a nota a atribuir, em cada um dos escalões, irá reflectir a estruturação da solução que deverá respeitar as normas da programação orientada aos objectos, cf aulas teóricas, nomeadamente o encapsulamento, a abstracção de implementação e a capacidade de a aplicação evoluir de forma controlada (eg: através da incorporação de novos tipos de artigos, transportadores, alteração das fórmulas, etc.).

Notem que **não serão considerados como merecedores de nota positiva trabalhos que não permitam a criação e visualização da informação relativa às entidades do programa.**

4 Relatório

O relatório deve descrever o trabalho realizado para desenvolver a aplicação solicitada. No mínimo, devem ser abordados os seguintes pontos:

- Capa com identificação da Unidade Curricular e do grupo (nome, número e fotografia).
- Descrição da arquitectura de classes utilizada (classes, atributos, etc.) e das decisões que foram tomadas na sua definição. Deverá ser entregue um **Diagrama de Classes** com a arquitectura de classes que suporta o programa desenvolvido.
- Descrição da aplicação desenvolvida (ilustração das funcionalidades).

Todos estes itens são obrigatórios!

5 Salvaguarda do estado da aplicação

O programa deve permitir que em qualquer momento se possa guardar em ficheiro a informação existente em memória sobre a informação relevante das entidades. A gravação deve ser feita de forma a permitir que o estado que foi gravado seja recuperado novamente. Na altura da entrega do projecto deve ser também entregue um estado (guardado em ficheiro) que possa ser carregado durante a apresentação.

6 Cronograma

1. Entrega da composição do grupo.

Data Limite: *18 de Abril de 2023* (esta fase é eliminatória, isto é, os grupos que não registarem a constituição não poderão submeter o projecto final).

2. Entrega final de código, ficheiro com os dados a carregar durante a apresentação e relatório de projecto (feita por via electrónica no elearning)

Data Entrega: *11 de maio de 2023* (podem entregar várias vezes o projecto, sendo que será apenas avaliada a última entrega)

3. Apresentação do projecto

Semana de: *15 a 19 de maio de 2023 - data acordada com a direcção de curso*