



Relatório do projeto

"Gestor de compras *online*"

Unidade Curricular: Programação Orientada aos Objetos

Docente responsável: Marília Curado

Curso: Licenciatura em Engenharia Informática

Filipe David Amado Mendes, nº 2020218797

9 de dezembro de 2021

Índice:

| | |
|-------------------------------|---|
| 1. Resumo | 3 |
| 2. Introdução | 3 |
| 3. Packages e classes | 4 |
| 4. Funcionamento | 5 |
| 5. Manual do utilizador | 7 |
| 6. Conclusão | 7 |

Resumo:

O projeto consiste na criação de uma aplicação em java que permita gerir as compras de um supermercado do ponto de vista da empresa, gerindo a base de dados (ficheiros de texto incluídos), mas principalmente do ponto de vista do utilizador. A aplicação deve permitir ao utilizador criar uma conta, fazer login com o seu email, ver as compras realizadas anteriormente e comprar produtos. Para tal são fornecidos três ficheiros de texto, "Clients.txt", "Products.txt" e "Purchase.txt", que contêm informação sobre clientes, produtos e compras e que são lidos pelo programa, facilitando assim a alteração de dados. Para que os dados sejam guardados permanentemente, o programa escreve-os num ficheiro de objetos que posteriormente substitui os ficheiros de texto e passa a ser lido a cada execução da aplicação.

Introdução:

Este projeto tem como objetivo avaliar as capacidades de interpretação, trabalho de equipa e programação orientada aos objetos dos alunos. Para tal a aplicação escrita em java é um trabalho complexo que tem como objetivo gerir as compras de um supermercado online, sendo a interação com o utilizador feita através do terminal.

Para a criação desta aplicação foi usado o editor de texto VSCode e alguns Plugins que ajudaram, por exemplo, na criação do *javadoc*.

Packages e Classes:

O programa está dividido em vários *packages* que se encontram dentro de uma diretoria src. As classes estão distribuídas pelos vários *packages* em função do seu papel na aplicação.

Os *packages* usados por este programa e as suas respectivas classes são:

1. Main.java
2. client
 - a. Client.java
 - b. Purchase.java
3. database
 - a. DataBaseManager.java
 - b. ReadFiles.java
4. date
 - a. Date.java
5. product
 - a. Product.java
 - b. Cleaning.java
 - c. Food.java
 - d. Furniture.java
6. promotion
 - a. Promotion.java
 - b. NoPromotion.java
 - c. PayLess.java
 - d. PaySomeItems.java
7. userinterface
 - a. UserInterface.java
 - b. FormatText.java

Funcionamento:

Este programa, faz uso de vários conceitos de programação orientada aos objetos, como herança, polimorfismo entre outros, tornando assim o uso de métodos mais eficiente e fácil.

Para tal, fazemos uso de *packages* que permitem ao programador agrupar classes semelhantes numa diretoria e importá-las mais tarde em outras classes, podendo assim criar objetos e fazer uso dos métodos das mesmas.

A class `Main.java` é a class principal do programa. Nela são criados objetos cruciais para a aplicação. É criado um objeto `DataBaseManager`, que permite gerir a base de dados do supermercado. É também criado um objeto `UserInterface`, que fará toda a parte de interação com o utilizador. É também importante realçar que é nesta class `Main`, no método `main()`, que são chamados os métodos que fazem a leitura dos ficheiros de texto e objetos.

No *package* `client` temos as classes que afetam os clientes. A class `Client.java` gere todos os atributos e métodos que manipulam os atributos dos objetos `Client`, como `showShoppingCart()`, que mostra o carrinho de compras do utilizador, `showPurchaseHistory()`, que mostra as compras realizadas pelo cliente anteriormente, ou até mesmo `acceptMbWayPayment()`, que verifica se as informações de MbWay foram bem introduzidas, verificando assim se o pagamento foi realizado com sucesso. Um método a destacar é o método `separateClientInfo()`, que recebe uma linha do ficheiro de texto "Clients.txt" e faz todo o parsing da informação, devolvendo um novo cliente que é depois adicionado a um `ArrayList`. A class `Purchase.java`, por outro lado, contém os atributos de uma compra e faz a ligação entre o cliente e o supermercado através do número de referência de uma compra.

No *package* `database`, temos todas as classes que gerem o supermercado. Desde leitura de ficheiros à manipulação de clientes e compras. A class `DataBaseManager.java` faz a gestão da base de dados do programa. Nela estão declarados como atributos três `ArrayLists`, um de clientes, um de produtos e um de compras. É através de um objeto desta classe que se manipula toda a base de dados, por exemplo, um cliente que crie uma nova conta ou faça uma compra. Um método relevante nesta classe é `createNewPurchase()`, que cria um novo objeto `Purchase`, permitindo assim adicioná-lo a lista `purchaseList`, que contém todas as compras do supermercado. A class `ReadFiles.java` trata de toda a leitura e escrita de ficheiros. Sendo a escrita apenas feita em ficheiros de objetos.

O *package* `date` apenas contém uma class. A class `Date.java`, faz a manipulação de datas. Permite que a aplicação use a data local do próprio computador ou que introduza manualmente uma data a sua escolha.

O *package* `product` tem todas as classes de produtos. A superclass `Product.java` tem definidos os atributos comuns a todos os produtos do supermercado. Além disso tem também implementações de métodos definidos também nas subclasses que tiram partido do polimorfismo.

As três subclasses de produtos, *Cleaning.java*, *Food.java* e *Furniture.java* têm os atributos de cada produto e métodos que afetam os respetivos produtos. Três métodos a destacar nestas classes são os métodos *separateProductInfo()*, que tiram partido do polimorfismo para fazer o parsing de toda a informação relativa aos produtos que e lida do ficheiro "Products.txt"

Visto que um supermercado aproveita promoções para conseguir aumentar os seus lucros, foi também aqui implementado um sistema de promoções. O *package* *promotion* contém quatro classes que gerem este sistema. As promoções são introduzidas pelo supermercado na base de dados (ficheiro "Products.txt") e os preços a elas associados são geridos pelas classes deste *package*. A classe *Promotion.java* é a superclass, que apenas tem definidos os atributos das promoções e um método *verifyPromotion()* que permite verificar se na data em que o cliente fez o *login* na aplicação as promoções se encontram disponíveis. A classe *NoPromotion.java* apenas indica que o produto não tem nenhuma promoção no momento. A classe *PaySomeItems.java* implementa uma promoção pague-três-leve-quatro. Por fim a classe *PayLess.java* implementa uma promoção pague-menos.

O último *package* utilizado é o *userinterface*, que trata de toda a parte visual e interativa do programa. A classe *UserInterface.java* tem apenas dois métodos: *getAppDate()* permite ao utilizador escolher o tipo de data que pretende utilizar, local ou introduzida manualmente; *menu()* faz toda a parte de interação com o utilizador e gestão do programa através de ciclos que permitem passar facilmente de um menu para o outro. Além disso, é responsável também por verificar se a escrita no ficheiro de objetos é feita sempre que alguma coisa é alterada pelo cliente, como uma nova compra ou alguma mudança no perfil. Por outro lado, a classe *FormatText.java* apenas trata da formatação de texto, como o alinhamento de *strings* no centro da consola.

Além de funcionalidades que proporcionam ao utilizador uma boa experiência, o programa tem também uma vasta gama de funcionalidades que permitem defender-lo contra erros ou uma má utilização da parte do cliente, sendo algumas delas:

- Verificação de existência de ficheiros e leitura do ficheiro adequado, isto é, caso não haja um ficheiro de objetos quando o programa inicia a leitura e feita nos ficheiros de texto
- Verificação de dados introduzidos pelo utilizador, não permitindo opções ou informações erradas.
- Verificação da escrita dos ficheiros de objetos.
- Entre outras.

Manual do utilizador:

O utilizador deverá correr o programa no terminal. Para isso deverá:

1. Certificar-se inicialmente que todos os *packages* e as respetivas classes estão numa diretoria `src/` que por sua vez se encontra na mesma diretoria dos ficheiros de texto "Clients.txt", "Products.txt" e "Purchases.txt".
2. Compilar o programa fazendo, usando um IDE a escolha, ou na consola, com o comando :

```
javac -d build src/*.java src/client/*.java src/database/*.java src/date/*.java src/product/*.java src/promotion/*.java src/userinterface/*.java
```

que compilará e colocará os ficheiros compilados numa diretoria `build/` .

3. Por último deverá correr o programa. Para tal é possível também o uso de um IDE, que faz uso de um terminal integrado, ou pelo terminal, com o comando:

```
java -cp build Main
```

que irá correr o programar a partir da diretoria `build/` .

Conclusão:

Concluindo assim este projeto, gostaria de expor a minha opinião quanto ao tema e a avaliação do projeto, à minha abordagem do mesmo e a possíveis futuros projetos.

O tema do projeto foi bastante interessante e dava muitas possibilidades quanto à implementação do programa. Porém, pelo facto de o trabalho ter sido pelos docentes como sendo a pares, acabei por sair bastante prejudicado, visto que realizei o projeto todo sozinho. Ainda assim, penso seriamente em fazer algum projeto deste género de novo, talvez até com o uso de interfaces gráficas.

Sendo assim, penso que abordei o projeto de uma forma criativa e eficaz e aprendi bastante com a realização deste projeto. Penso que o programa tem uma boa robustez e uma interface muito boa, seguindo os bons princípios de programação orientada aos objetos.