|  |  |
| --- | --- |
| ERP na nuvem: Mitos e verdades que você precisa conhecer | Cake ERP  Gestor de compras online  Projeto Final | PROGRAMAÇÃO ORIENTADA AOS OBJETOS |



Departamento de Engenharia Informática

Faculdade De Ciências e Tecnologias

Universidade de Coimbra

Ano Letivo 2021/2022

Ana Rita Martins Oliveira

2020213684

Mariana Ferreira Sousa

2020226346

Índice

[Resumo 2](#_Toc89985750)

[Introdução 3](#_Toc89985751)

[Descrição das classes e principais métodos 4](#_Toc89985752)

[Estrutura dos ficheiros de texto 6](#_Toc89985753)

[Manual de utilizador 7](#_Toc89985754)

[Diagrama de classes 9](#_Toc89985755)

[Conclusão 10](#_Toc89985756)

[Referências 11](#_Toc89985757)

# Resumo

De um modo geral, foi-nos proposto o desenvolvimento de uma aplicação para gerir compras online. A aplicação será implementada na linguagem Java e este trabalho abordará os seguintes aspetos:

* Elaboração de um diagrama de classes (UML);
* Elaboração do código da aplicação em java;
* Elaboração do *Javadoc*.

# Introdução

O presente trabalho aborda uma cadeia de supermercados que pretende desenvolver uma aplicação para a venda online dos seus produtos, sendo que o objetivo principal do mesmo é conseguir gerir clientes, produtos, vendas e promoções.

Visto isto, desenvolvemos uma aplicação, para gerir o mesmo e em que a interação com o utilizador se efetuará através da consola, ou seja, vai aparecendo um menu com várias opções em que o utilizador escolherá a que pretende ao longo da aplicação.

Nas páginas seguintes estarão descritas as principais classes e métodos bem como as suas funcionalidades. Ainda se encontrará explicada qual a estrutura a utilizar nos ficheiros de texto de modo a permitir a fácil e rápida edição dos mesmos, um manual do utilizador (como o utilizador deve correr a aplicação de forma correta) e um diagrama de classes.

# Descrição das classes e principais métodos

A aplicação desenvolvida para este projeto apresenta diversas Classes e Métodos importantes. Sendo a mesma desenvolvida emJava a relação entre estas encontram-se apresentadas de forma mais explicita ao nível do UML associado a este documento.

Contudo, é agora apresentada uma breve explicação associada a cada uma das Classes.

* GerirSupermercado: Classe onde se encontra o método main, são apresentados os menus da aplicação, confirmados os dados de *login,* registado um novo cliente e é concretizada a compra.
* Ficheiro: Classe onde se encontram os métodos de leitura em ficheiros de objetos de texto e objetos e o de escrita em ficheiro de objetos.
* Supermercado: Classe onde são guardados e podem ser consultados os cliente, os produtos e as promoções do supermercado e onde se encontram os métodos de acesso a produtos;
* Cliente: Classe onde se encontram os métodos onde são guardadas e podem ser consultadas as informações pessoais acerca dos clientes, entre elas as compras por ele já realizadas;
* venda: Classe onde se encontram os métodos onde são guardadas e podem ser consultadas as informações acerca de cada venda e o método que aplica o desconto a cada a cada compra com promoção:
* produto: Classe abstrata onde temos os métodos onde são guardadas e podem ser consultadas as caracterizações acerca de cada produto, é inicializado o método abstrato peso e o método escreve que será utilizado também pelas classes descendentes;
* alimentar: Classe descendente da classe Produto, onde ainda são guardados e podem ser consultados o nº de calorias por 100g e a percentagem de gordura do produto, o método abstrato peso é implementado retornando o valor 0, pois a característica peso não é um atributo desta classe, o método escreve é chamado usando o polimorfismo e mostra no ecrã todas as características do alimento;
* limpeza: Classe descendente da classe Produto, onde ainda é guardado e pode ser consultado o grau de toxicidade (escala de 0 a 10), o método abstrato peso é implementado retornando o valor 0, pois a característica peso não é um atributo desta classe, o método escreve é chamado usando o polimorfismo e mostra no ecrã todas as características do produto de limpeza;
* mobiliario: Classe descendente da classe Produto, onde ainda são guardados e podem ser consultados o peso e a dimensão, o método abstrato peso é implementado retornando o valor do peso do produto, que será usado para se saber se existe custo adicional de transporte ou não, o método escreve é chamado usando o polimorfismo e mostra no ecrã todas as características do produto de mobiliário;
* Dimensaoo: Classe para formar as dimensões dos produtos de mobiliário com largura, altura e profundidade;
* data: Classe onde temos os métodos onde são guardadas e podem ser consultadas as caracterizações acerca de cada data e os métodos de acesso às datas;
* promocao: Classe abstrata onde temos os métodos onde são guardadas e podem ser consultadas as caracterizações acerca de cada promoção e é inicializado o método abstrato desconto;
* TresQuatro: Classe descendente da classe Promoção onde é implementado o método abstrato desconto que retorna o valor do desconto que se tem de aplicar à compra final, aquando da aplicação da promoção pague-três-leve-quatro.
* menos: Classe descendente da classe Promoção onde é implementado o método abstrato desconto que retorna o valor do desconto que se tem de aplicar à compra final, aquando da aplicação da promoção pague-menos.

# Estrutura dos ficheiros de texto

* Ficheiros de texto

O método que utilizamos para estruturar os nossos ficheiros de texto foi separar cada parcela correspondente a cada atributo de cada classe por vírgulas, caso esse atributo seja uma data separemos cada atributo da data por uma barra (/) e caso seja uma dimensão com um ‘x’ a separar cada número correspondente aos atributos da dimensão.

# Manual de utilizador

O programa interage com o utilizador através da consola, onde são pedidos os dados precisos do mesmo.

Uma imagem com texto

Descrição gerada automaticamente

Menu 1

O primeiro passo é escolher se se quer realizar o *login* carregando em 1 ou fazer um novo registo, carregando em 2, caso o cliente ainda não tenha ficha no supermercado.

Carregando em 1:

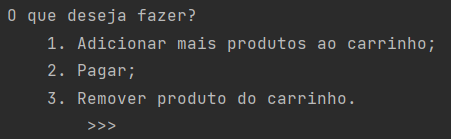
Uma imagem com texto

Descrição gerada automaticamente

Menu 2

É pedido ao utilizador que insira o seu e-mail e caso seja válido, isto é, caso já exista na base de dados do supermercado, o login é efetuado com sucesso e é perguntado ao utilizador se deseja realizar uma compra, carregando em 1, consultar as compras realizadas, carregando em 2 ou fazer *logout* carregando em 3.

De seguida, é pedido ao utilizador que insira a data da compra e são listados todos os produtos do supermercado, tendo depois o utilizador que inserir o identificador do produto que deseja comprar e a respetiva quantidade.



Menu 3

Depois de adicionar um produto à sua compra o utilizador pode escolher adicionar mais produtos ao carrinho, carregando em 1, e nesse caso irá voltar a aparecer a lista dos produtos do supermercado e será pedido o número de quantidades que quer comprar. Caso carregue em 2 irá partir para o pagamento e caso carregue em 3 remove produtos do carrinho.

Se carregar em 2 é apresentada uma fatura e de seguida o seguinte menu 4, se carregar em 3 é logo apresentado o menu 4.

Uma imagem com texto

Descrição gerada automaticamente

Menu 4

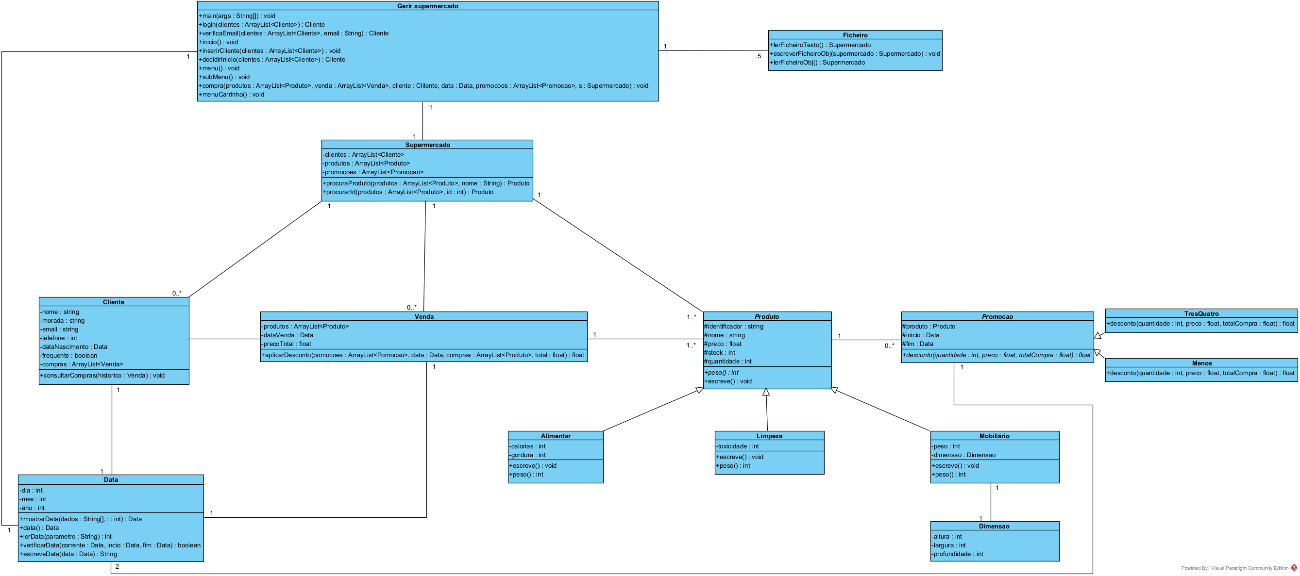
Quando o utilizador carrega em 1 começa a efetuar nova compra, pedindo um nove e-mail, caso carregue em 2 sai da aplicação.

No menu 2, se se carregar em 2 são listadas todas as compras já efetuadas pelo cliente e em 3 é feito o *logout*.

Carregando em 2 no menu 1, é possível registar um novo cliente e de seguida realizar todas as opções do menu 2.

# Diagrama de classes

O Diagrama de Classes encontra-se apresentado abaixo, contudo e de forma a ser mais percetível as relações entre as mesmas, este estará disponibilizado em pdf com os restantes elementos do trabalho.



# Conclusão

Este Projeto Final permitiu-nos pôr em prática os conhecimentos adquiridos nas aulas Teórico-Práticas da cadeira de Programação Orientada aos Objetos. Com ele aprendemos mais sobre os conteúdos lecionado, assim como sobre a Linguagem *Java*.

Tal como referido na introdução, o objetivo deste trabalho prático baseava-se principalmente na utilização correta dos conceitos herança, Polimorfismo, Ficheiros, entre outros, e ainda manipular de forma eficaz objetos, bem como aplicar o conceito de *ArrayList*.

Por fim, as noções abstratas desta cadeira estão, agora, mais claras.

# Referências

* Material disponibilizado pelo docente da cadeira de Programação Orientada aos Objetos.